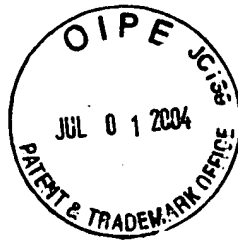


00862.023463.



IFW

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)	
	:	Examiner: Not Yet Assigned
KITAHIRO KANEDA ET AL.)	
	:	Group Art Unit: Not Yet Assigned
Application No.: 10/781,869)	
	:	
Filed: February 20, 2004)	
	:	
For: IMAGE PROCESSING METHOD)	
AND IMAGE PROCESSING	:	
SYSTEM)	June 30, 2004

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

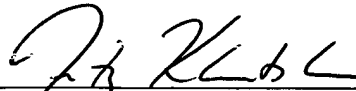
Sir:

In support of Applicants' claim for priority under 35 U.S.C. § 119, enclosed is
a certified copy of the following foreign application:

JAPAN 2003-044299, filed February 21, 2003.

Applicants' undersigned attorney may be reached in our New York office by telephone at (212) 218-2100. All correspondence should continue to be directed to our address given below.

Respectfully submitted,



Attorney for Applicants
Fritz Klantschi

Registration No. 50,333

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3800
Facsimile: (212) 218-2200

NY_MAIN 437440v1

CFM 03463

US

CN

00862.023463

10/781,869

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 2 月 2 1 日
Date of Application:

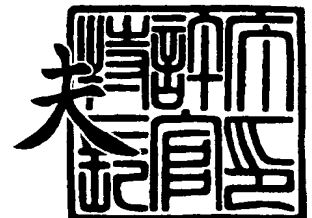
出 願 番 号 特 願 2 0 0 3 - 0 4 4 2 9 9
Application Number:
[ST. 10/C] : [J P 2 0 0 3 - 0 4 4 2 9 9]

出 願 人 キヤノン株式会社
Applicant(s):

2 0 0 4 年 3 月 8 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 4 - 3 0 1 7 6 9 8

【書類名】 特許願

【整理番号】 253364

【提出日】 平成15年 2月21日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 G06T 1/00

【発明の名称】 画像処理方法

【請求項の数】 1

【発明者】

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社
内

【氏名】 金田 北洋

【発明者】

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社
内

【氏名】 金津 知俊

【発明者】

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社
内

【氏名】 三沢 玲司

【発明者】

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社
内

【氏名】 伊藤 裕彦

【発明者】

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社
内

【氏名】 秋庭 朋宏

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号キャノン株式会社
内

【氏名】 寺尾 仁秀

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号キャノン株式会社
内

【氏名】 鶴沢 充

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号キャノン株式会社
内

【氏名】 加藤 進一

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号キャノン株式会社
内

【氏名】 谷岡 宏

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号キャノン株式会社
内

【氏名】 宇佐美 彰浩

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号キャノン株式会社
内

【氏名】 太田 健一

【特許出願人】

【識別番号】 000001007

【住所又は居所】 東京都大田区下丸子3丁目30番2号

【氏名又は名称】 キャノン株式会社

【代表者】 御手洗 富士夫

【代理人】**【識別番号】** 100090538**【住所又は居所】** 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社
内**【弁理士】****【氏名又は名称】** 西山 恵三**【電話番号】** 03-3758-2111**【選任した代理人】****【識別番号】** 100096965**【住所又は居所】** 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会
社内**【弁理士】****【氏名又は名称】** 内尾 裕一**【電話番号】** 03-3758-2111**【手数料の表示】****【予納台帳番号】** 011224**【納付金額】** 21,000円**【提出物件の目録】****【物件名】** 明細書 1**【物件名】** 図面 1**【物件名】** 要約書 1**【包括委任状番号】** 9908388**【プルーフの要否】** 要

【書類名】 明細書

【発明の名称】 画像処理方法

【特許請求の範囲】

【請求項1】 入力されたイメージそれぞれに対して、格納手段に格納されているオリジナル電子データを検索する検索ステップと、

前記検索ステップでオリジナル電子データが検索されなかった場合、前記入力されたイメージをベクトルデータ化するベクトル化ステップと、

前記ベクトル化ステップでベクトルデータ化されたイメージを電子データとして格納手段に格納する格納ステップと、

前記検索ステップでオリジナル電子データが検索された場合において当該検索されたオリジナル電子データに関する情報、あるいは前記検索ステップでオリジナル電子データが検索されなかった場合において前記ベクトル化ステップでベクトル化されて前記格納ステップで格納された電子データに関する情報、の少なくともいずれかを含むシートを生成するシート生成ステップとを有することを特徴とする画像処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本願発明は、複写機などの画像処理装置で読み取った画像データを、Word等所謂文書作成アプリケーションソフトで再利用可能なベクトルデータに変換し、それらを簡便に利用する画像処理方法に関する。

【0002】

【従来の技術】

近年、環境問題が叫ばれる中、オフィスでのペーパーレス化が急速に進んでおり、電子文書を取り扱う各種技術が考え出されている。

【0003】

例えば、特許文献1では、紙文書をスキャナーで読み取り、電子文書フォーマット（JPEGやPDFなど）に変換して、画像記憶手段に格納させるものが記載されている。

【0 0 0 4】

また、特許文献 2 では、文書画像内に含まれる属性毎の領域を検出し、該領域毎にコンテンツとして、文書を管理する文書管理システムが記載されている。

【0 0 0 5】

また、特許文献 3 では、デジタル複合機（コピー機能、スキャン機能、プリント機能などを備える）において、スキャンしたイメージ中にページ I D を示すグラフィック・コードがあるかどうか確認し、グラフィック・コードが見つければ、該当するページ I D をデータベース上で検索する。そして、もし、データベース上でページ I D が見つければ、今取り込んだばかりのスキャン・イメージを廃棄し、これに代わってページ I D に関連付けられた印刷データを取り出して、プリント操作によって印刷イメージを生成し且つ用紙に印刷する技術が記載されている。一方、データベース上で該当するページ I D が見つからなかった場合、コピー操作時はスキャン・イメージをそのまま用紙に複写し、ファクシミリ操作時やファイリング操作時は、スキャン・イメージに P D L コマンドを付与して P D L フォーマットにして送信することが記載されている。

【0 0 0 6】

【特許文献 1】

特開 2 0 0 1 - 3 5 8 8 6 3 号公報

【特許文献 2】

特開平 8 - 1 4 7 4 4 5 号公報

【特許文献 3】

特開平 1 0 - 2 8 5 3 7 8 号公報

【0 0 0 7】

【発明が解決しようとする課題】

しかしながら、特許文献 1 の技術では、スキャナーで読み取った画像をコンパクトな情報量の J P E G ファイルや P D F ファイルとして保存可能であるが、プリントされた文書から保存されているファイルを検索することができないので、プリントとスキャンを繰り返すと、保存している文書の画像が劣化していつてしまうという課題がある。

【0008】

また、特許文献2の技術では、画像を複数の領域に分割してコンテンツ毎に再利用可能としているが、ユーザの指示に基づいて、コンテンツを検索し、更にもの中から使用するコンテンツを決めるため、格納されているコンテンツを利用して文書を作成する際、ユーザはどのコンテンツを用いるか決めなければならず非常に手間がかかってしまうという課題がある。

【0009】

また、特許文献3の技術では、紙文書に対応するオリジナルの電子文書が見つからなかった場合にスキャン・イメージにPDLコマンドを付与してPDLフォーマットにしているが、単にイメージにPDLコマンドを付与してPDLフォーマットにするだけではファイルサイズが比較的大きくなってしまいう課題がある。

【0010】**【課題を解決するための手段】**

上記目的を達成するために、本発明の画像処理方法は、入力されたイメージそれぞれに対して、格納手段に格納されているオリジナル電子データを検索する検索ステップと、前記検索ステップでオリジナル電子データが検索されなかった場合、前記入力されたイメージをベクトルデータ化するベクトル化ステップと、前記ベクトル化ステップでベクトルデータ化されたイメージを電子データとして格納手段に格納する格納ステップと、前記検索ステップでオリジナル電子データが検索された場合において当該検索されたオリジナル電子データに関する情報あるいは前記検索ステップでオリジナル電子データが検索されなかった場合において前記ベクトル化ステップでベクトル化されて前記格納ステップで格納された電子データに関する情報の少なくともいずれかを含むシートを生成するシート生成ステップとを有することを特徴とする。

【0011】**【発明の実施の形態】**

本願発明の実施の形態について説明する。図1は本願発明にかかる画像処理システム構成例を示すブロック図である。この画像処理システムは、オフィス10

とオフィス 20 とをインターネット 104 で接続された環境で実現する。オフィス 10 内に構築された LAN 107 には、MFP (Multi Function Peripheral、デジタル複合機) 100、該 MFP 100 を制御するマネジメント PC 101、クライアント PC 102、文書管理サーバ 106、そのデータベース 105、およびプロキシサーバ 103 が接続されている。LAN 107 及びオフィス 20 内の LAN 108 はプロキシサーバ 103 を介してインターネット 104 に接続される。MFP 100 は本発明において紙文書の画像読み取り処理 (スキャナー) と読み取った画像信号に対する画像処理の一部を担当し、画像信号は LAN 109 を用いてマネジメント PC 101 に入力する。マネジメント PC は通常の PC であり、内部に画像記憶手段、画像処理手段、表示手段、入力手段を有するが、その一部は MFP 100 に一体化されて構成されているものとする。

【0012】

図 2 は MFP 100 の構成図である。図 2 においてオートドキュメントフィーダー (以降 ADF と記す) を含む画像読み取り部 110 は束状の或いは 1 枚の原稿画像を図示しない光源で照射し、原稿反射像をレンズで固体撮像素子上に結像し、固体撮像素子からラスタ状の画像読み取り信号を例えば 600 DPI の密度のイメージ情報として得る。通常の複写機能はこの画像信号をデータ処理部 115 で記録信号へ画像処理し、複数毎複写の場合は記録装置 111 に一旦記録データを記憶保持した後、記録装置 112 に順次出力して紙上に画像を形成する。

【0013】

一方クライアント PC 102 から出力されるプリントデータは LAN 107 からネットワーク IF 114 を経てデータ処理装置 115 で記録可能なラスタデータに変換した後、前記記録装置で紙上に記録画像として形成される。

【0014】

MFP 100 への操作者の指示は、MFP に装備されたキー操作部とマネジメント PC に入力されるキーボード、及び、マウスからなる入力装置 113 から行われ、これら一連の動作はデータ処理装置 115 内の図示しない制御部で制御される。

【0015】

一方、操作入力の状態表示及び処理中の画像データの表示は表示装置 116 で行われる。尚記憶装置 111 はマネージメント PC から制御され、これら MFP とマネージメント PC とのデータの授受及び制御はネットワーク IF 117 および直結した LAN 109 を用いて行われる。

【0016】

なお、本発明は、図 2 に示した装置ないし図 1 に示したシステムにおいて実行されうるものであり、例えば、図 2 の記憶装置 111 に記憶されたコンピュータ実行可能な制御プログラム（なお、この場合、本発明を構成する制御プログラムは記憶装置 111 に記憶されていてもいいし、外部装置から通信回線等を介して読み込まれて実行されるものであってもよい。また記憶装置は内蔵型ハードディスクに限らずリムーバブルディスクなどであってもよい）を、データ処理装置 115 の制御部（CPU）が実行することにより、後述する本発明の処理が実行されるものであってもいいし、また、前記データ処理装置 115 の制御部の全部又は一部が電気回路のハードウェアで構成され該電気回路で信号を処理することにより本発明の処理を実行するものであってもよい。

【0017】

<<処理概要>>

次に本発明による画像処理全体の概要を図 3 を用いて説明する。

【0018】

図 3 においてまず、MFP 100 の画像読み取り部 110 を動作させ 1 枚の原稿をラスタ状に走査し、イメージ情報入力処理 120 で 600 DPI-8 ビットの画像信号を得る。該画像信号をデータ処理部 115 で前処理を施し記憶装置 111 に 1 ページ分の画像データとして保存する。マネージメント PC 101 の CPU は該格納された画像信号から先ず、文字／線画部分とハーフトーンの画像部分とに領域を分離し、文字部は更に段落で塊として纏まっているブロック毎に分離し、また、線画部分は線で構成された表、図形毎に分離し、各々セグメント化する。一方、ハーフトーンで表現される画像部分は、矩形に分離されたブロックの画像部分、背景部等、所謂ブロック毎に独立したオブジェクトに分割する（

ステップ121)。

【0019】

このとき原稿画像中に付加情報として記録された2次元バーコード、或いはURLに該当するオブジェクトを検出し、URLについては文字認識処理(OCR)を実行し、或いは2次元バーコードなら該マークを解読(OMR)して(ステップ122)、該原稿のオリジナル電子ファイルが格納されている記憶装置内のポインター情報を検出する(ステップ123)。尚、ポインター情報を付加する手法としては、他に文字と文字の間隔を微小に変化させることによって情報を埋め込む手法や、ハーフトーンの画像に電子透かしとして埋め込む手法等であってもよい。

【0020】

ポインター情報が検出された場合、ステップ125に分岐し、ポインターで示されたアドレスから元の電子ファイルを検索する。電子ファイルは図1においてクライアントPC内のハードディスク内、或いはオフィス10或いは20のLANに接続された文書管理サーバ106によって管理されるデータベース105内、或いはMFP100自体が有する記憶装置111のいずれかに格納されており、ステップ123で得られたアドレス情報に従ってこれらの記憶装置内を検索する。ステップ125で電子ファイルが見つからなかった場合、見つかったがJPEGやPDFあるいはtiffに代表される所謂イメージファイルであった場合、或いは、ステップ124でポインター情報自体が存在しなかった場合は、ステップ126に分岐する。125でポインター情報に基づく電子ファイルが見つかった場合、ステップ133へ分岐する。

【0021】

ステップ126は所謂文書検索処理ルーチンである。まずステップ122で各文字ブロックに対して行ったOCRの結果から単語を抽出して電子ファイル内に含まれる単語と比較することにより全文検索、或いは各オブジェクトの配列と各オブジェクトの属性を電子ファイルのオブジェクト配列及びオブジェクト属性と比較することにより所謂レイアウト検索を行う。検索の結果、類似度の高い電子ファイルが見つかった場合、サムネイル等で候補の電子ファイルを表示(ステッ

プ 1 2 7) し、複数の候補中から操作者の選択が必要なら操作者の入力操作によってファイルの特定を行う。電子ファイルが特定された場合はステップ 1 3 3 に進むが、ステップ 1 2 6 の検索処理で電子ファイルが見つからなかった場合、或いは、特定された電子ファイルが P D F あるいは t i f f に代表される所謂イメージファイルであった場合、ステップ 1 2 9 に分岐する。

【 0 0 2 2 】

ステップ 1 2 9 はイメージデータからベクトルデータへの変換処理部であり、イメージ情報をベクトル化した電子ファイルに変換する。まず、ステップ 1 2 2 で O C R された文字ブロックに対しては、更に文字のサイズ、スタイル、字体を認識し、原稿を走査して得られた文字に可視的に忠実なフォントデータに変換する。一方、線で構成される表、図形ブロックに対してはアウトライン化する。写真などの自然画像ブロックに対してはイメージデータとして個別の J P E G ファイルとして処理する。これらのベクトル化処理は各オブジェクト毎に行い、更に各オブジェクトのレイアウト情報を保存して、ステップ 1 3 0 で一般のアプリケーションで編集可能なアプリデータに変換され、ステップ 1 3 1 で電子ファイルとして記憶装置 1 1 1 に格納され、更に、以降同様の処理を行う際に直接電子ファイルとして検索出来るように、ステップ 1 3 2 において検索の為のインデックス情報を生成して検索用インデックスファイルに追加される。なお、ステップ 1 3 0 で、汎用のファイル形式として、例えば、r t f (R i c h T e x t F o r m a t) 形式に変換するようにしてもよい。

【 0 0 2 3 】

以上の処理が終了した後、ステップ 1 3 3 で当該ページが最終ページか否かを判別し、そうでない場合は、ステップ 1 2 0 に制御を戻し、次のページの処理を再び繰り返す。

【 0 0 2 4 】

最終ページであった場合は、ステップ 1 3 4 に制御を移行する。

【 0 0 2 5 】

ステップ 1 3 4 では、本発明の主眼を為す処理で、上記、スキャン、検索、ベクトル化を経て特定された全ページのページ情報（例えば、サムネイル、タイト

ル、要約、キーワード、ID、日付、著作者、及び処理結果へのポインター情報など）を含むインデックスシート（以後ポータルシートと呼ぶことにする）を生成し、出力する。

【0026】

以下、各処理ブロックに対して詳細に説明する。

【0027】

<<ブロックセレクション処理>>

先ずステップ121で示すブロックセレクション処理について説明する。

【0028】

ブロックセレクション処理とは、図4の右に示すように、ステップ120で入力されたイメージデータ（図4左）を、各オブジェクト毎の塊として認識し、該ブロック各々を文字／図画／写真／線／表等の属性に判定し、異なる属性を持つ領域に分割する処理である。

【0029】

ブロックセレクション処理の実施例を以下に説明する。

【0030】

先ず、入力画像を白黒に二値化し、黒画素の輪郭線追跡をおこなって、黒画素輪郭で囲まれる画素の塊を抽出する。面積の大きい黒画素の塊については、内部にある白画素に対しても輪郭線追跡をおこない白画素の塊を抽出、さらに一定面積以上の白画素の塊の内部からは再帰的に黒画素の塊を抽出する。

【0031】

このようにして得られた黒画素の塊を、大きさおよび形状で分類し、異なる属性を持つ領域へ分類していく。たとえば、縦横比が1に近く、大きさが一定の範囲のものを文字相当の画素塊として、さらに近接する文字の画素塊が整列良くグループ化可能な部分を文字領域とし、また、扁平な画素塊を線領域とし、一定大きさ以上でかつ四角系の白画素塊を整列よく内包する黒画素塊の占める範囲を表領域とし、不定形の画素塊が散在している領域を写真領域とし、それ以外の任意形状の画素塊を図画領域、などとする。

【0032】

ブロックセレクション処理で得られた各ブロックに対するブロック情報と、入力画像に含まれるブロックを管理するための入力ファイル情報とを図5に示す。これらのブロック毎の情報は以降に説明するベクトル化、或いは検索の為の情報として用いる。

【0033】

＜＜ポインター情報の検出＞＞

次に、ファイルの格納位置をイメージ情報から抽出する為のOCR/OMR処理（ステップ122）について説明する。

【0034】

図6は原稿画像中に付加された2次元バーコード（QRコードシンボル）を復号して、データ文字列を出力する過程を示すフローチャートである。2次元バーコードの付加された原稿310の一例を図7に示す。

【0035】

まず、データ処理装置115内のページメモリに格納された原稿310を表すイメージ画像をCPU（不図示）で走査して、先に説明したブロックセレクション処理の結果から、所定の2次元バーコードシンボル311の位置を検出する。QRコードの位置検出パターンは、シンボルの4隅のうちの3隅に配置される同一の位置検出要素パターンから構成される（ステップ300）。

【0036】

次に、位置検出パターンに隣接する形式情報を復元し、シンボルに適用されている誤り訂正レベルおよびマスクパターンを得る（ステップ301）。

【0037】

シンボルの型番を決定した（ステップ302）後、形式情報で得られたマスクパターンを使って符号化領域ビットパターンをXOR演算することによってマスク処理を解除する（ステップ303）。

【0038】

尚、モデルに対応する配置規則に従い、シンボルキャラクタを読み取り、メッセージのデータ及び誤り訂正コード語を復元する（ステップ304）。

【0039】

復元されたコード上に、誤りがあるかどうかの検出を行い（ステップ 3 0 5）、誤りが検出された場合、ステップ 3 0 6 に分岐し、これを訂正する。

【 0 0 4 0 】

誤り訂正されたデータより、モード指示子および文字数指示子に基づいて、データコード語をセグメントに分割する（ステップ 3 0 7）。

【 0 0 4 1 】

最後に、仕様モードに基づいてデータ文字を復号し、結果を出力する（ステップ 3 0 8）。

【 0 0 4 2 】

尚、2 次元バーコード内に組み込まれたデータは、対応するファイルのアドレス情報を表しており、例えばファイルサーバ名およびファイル名からなるパス情報で構成される。或いは、対応するファイルへの URL で構成されるものであってもよい。

【 0 0 4 3 】

本実施例ではポインター情報が 2 次元バーコードを用いて付与された原稿 3 1 0 について説明したが、文字列でポインター情報が記録されるようにしてもよい。その場合は所定のルールに従った文字列のブロック（例えば予め決められた位置にある文字ブロック）を先のブロックセレクション処理で検出し、該ポインター情報を示す文字列ブロックの各文字を文字認識する事で、元ファイルのアドレス情報を得る事が可能である。

【 0 0 4 4 】

又、図 7 の文書 3 1 0 の文字ブロック 3 1 2、或いは 3 1 3 の文字列に対して隣接する文字と文字の間隔等に微小な変調を加えるなどすることにより、該文字間隔に情報を埋め込むことでもポインター情報を付与できるので、例えば、後述する文字認識処理を行う際に各文字の間隔を検出すれば、ポインター情報が得られる。又、自然画 3 1 4 の中に不可視の電子透かしとしてポインター情報を付加する事も可能である。

【 0 0 4 5 】

<<ポインター情報によるファイル検索>>

次に、図3のステップ125で示した、ポインタ情報からの電子ファイルの検索について、図8のフローチャートを使用して説明する。

【0046】

まず、ポインタ情報に含まれるアドレスに基づいて、ファイル・サーバーを特定する（ステップ400）。

【0047】

ここでファイルサーバとは、クライアントPC102や、データベース105を内蔵する文書管理サーバ106や、記憶装置111を内蔵するMF P100自身を指す。また、ここでアドレスとは、URLや、サーバ名とファイル名からなるパス情報である。

【0048】

ファイルサーバが特定できたら、ファイルサーバに対してアドレスを転送する（ステップ401）。ファイル・サーバは、アドレスを受信すると、該当するファイルを検索する（ステップ402）。ファイルが存在しない場合（ステップ403-N）には、MF Pに対してその旨通知する。

【0049】

一方、ファイルが存在した場合（ステップ403-Y）には、ファイルのアドレスを通知し、ステップ134の処理で用いるようにする（ステップ408）。

【0050】

<<ファイル検索処理>>

次に、図3のステップ126で示すファイル検索処理の詳細について図5、図10を使用して説明を行う。

【0051】

ステップ126の処理は、前述したように、ステップ124で入力原稿（入力ファイル）にポインタ情報が存在しなかった場合、または、ステップ125でポインタ情報は在るが電子ファイルが見つからなかった場合、或いは電子ファイルがイメージファイルであった場合に行われる。

【0052】

ここでは、ステップ122の結果、抽出された各ブロック及び入力ファイルが

、図5に示す情報（ブロック情報、入力ファイル情報）を備えるものとする。情報内容として、属性、座標位置、幅と高さのサイズ、OCR情報有無を例としてあげる。属性は、文字、線、写真、絵、表、その他に分類する。また簡単に説明を行うため、ブロックは座標Xの小さい順、即ち（例、 $X1 < X2 < X3 < X4 < X5 < X6$ ）にブロック1、ブロック2、ブロック3、ブロック4、ブロック5、ブロック6と名前をつけている。ブロック総数は、入力ファイル中の全ブロック数であり、図10の場合は、ブロック総数は6である。以下、これらの情報を使用して、データベース内から、入力ファイルに類似したファイルのレイアウト検索を行うフローチャートを図10に示す。ここで、データベースに格納されているファイルは、図5と同様の種類の情報（ブロック情報、ファイル情報）を備えることを前提とする。

【0053】

フローチャートの流れは、入力ファイルとデータベース中のファイルを順次比較するものである。まず、ステップ510にて、後述する類似率などの初期化を行う。次に、ステップ511にてブロック総数の比較を行い、ここで、真の場合、さらにファイル内のブロックの情報を順次比較する。ブロックの情報比較では、ステップ513、515、518にて、属性類似率、サイズ類似率、OCR類似率をそれぞれ算出し、ステップ522にてそれらをもとに総合類似率を算出する。各類似率の算出方法については、公知の技術を用いることができるので説明を省略する。ステップ523にて総合類似率が、予め設定された閾値 T_h より高ければステップ524にてそのファイルを類似候補としてあげる。但し、図中の N 、 W 、 H は、入力ファイルのブロック総数、各ブロック幅、各ブロック高さとし、 ΔN 、 ΔW 、 ΔH は、入力ファイルのブロック情報を基準として誤差を考慮したものである。 n 、 w 、 h は、データベースに格納されているファイルのブロック総数、各ブロック幅、各ブロック高さとする。また、不図示ではあるが、ステップ514にてサイズ比較時に、位置情報 XY の比較などを行ってもよい。

【0054】

以上、検索の結果、類似度が閾値 T_h より高く、候補として保存されたデータベースファイル（ステップ524）をサムネイル等で表示（ステップ127）す

る。そして、複数の候補が表示された場合、この中から操作者の入力操作によってファイルの特定を行う。

【0 0 5 5】

<<ベクトル化処理>>

次に、図3のステップ129で示されるベクトル化処理について詳説する。ここでは、ファイルサーバにオリジナル電子ファイルが存在しないと判断された場合に、図4に示すイメージデータ（スキャン入力されたイメージデータ）を各ブロック毎にベクトル化していく。

【0 0 5 6】

<文字ブロックのベクトル化>

まず、文字ブロックに対しては各文字に対して文字認識処理を行う。文字認識部では、文字単位で切り出された画像に対し、パターンマッチの一手法を用いて認識を行い、対応する文字コードを得る。この認識処理は、文字画像から得られる特徴を数十次元の数値列に変換した観測特徴ベクトルと、あらかじめ字種毎に求められている辞書特徴ベクトルと比較し、最も距離の近い字種を認識結果とする処理である。特徴ベクトルの抽出には種々の公知手法があり、たとえば、文字をメッシュ状に分割し、各メッシュ内の文字線を方向別に線素としてカウントしたメッシュ数次元ベクトルを特徴とする方法がある。

【0 0 5 7】

ブロックセレクション（ステップ121）で抽出された文字領域に対して文字認識を行う場合は、まず該当領域に対し横書き、縦書きの判定をおこない、各々対応する方向に行を切り出し、その後文字を切り出して文字画像を得る。横書き、縦書きの判定は、該当領域内で画素値に対する水平／垂直の射影を取り、水平射影の分散が大きい場合は横書き領域、垂直射影の分散が大きい場合は縦書き領域と判断すればよい。文字列および文字への分解は、横書きならば水平方向の射影を利用して行を切り出し、さらに切り出された行に対する垂直方向の射影から、文字を切り出すことでおこなう。縦書きの文字領域に対しては、水平と垂直を逆にすればよい。尚、この時文字のサイズが検出出来る。

【0 0 5 8】

次に、該文字ブロック内の各文字のフォント種別を認識する。文字認識の際に用いる、字種数ぶんの辞書特徴ベクトルを、文字形状種すなわちフォント種に対して複数用意し、マッチングの際に文字コードとともにフォント種を出力することで、文字のフォントが認識出来る。

【0059】

前記文字認識およびフォント認識によって得られた、文字コードおよびフォント情報を用いて、文字コード及びフォントごとにあらかじめ用意されたアウトラインデータを用いて、文字部分の情報をベクトルデータに変換する。なお、元原稿がカラーの場合は、カラー画像から各文字の色を抽出してベクトルデータとともに記録する。

【0060】

以上の処理により文字ブロックに属するイメージ情報をほぼ形状、大きさ、色が忠実なベクトルデータに変換出来る。

【0061】

＜線画ブロック、図画ブロック等のベクトル化＞

ブロックセレクション処理（ステップ121）で、線画、図画、あるいは表領域とされた領域を対象に、該領域中で抽出された画素塊の輪郭をベクトルデータに変換する。具体的には、輪郭をなす画素の点列を角と看做される点で区切って、各区間を部分的な直線あるいは曲線で近似する。角とは曲率が極大となる点であり、曲率が極大となる点は、図11に図示するように、任意点 P_i に対し左右 k 個の離れた点 P_{i-k} と P_{i+k} の間に弦を引いたとき、この弦と P_i の距離が極大となる点として求められる。さらに、（ P_{i-k} と P_{i+k} 間の弦の長さ／弧の長さ）を R とし、 R の値が閾値以下である点を角とみなすことができる。角によって分割された後の各区間は、直線は点列に対する最小二乗法など、曲線は3次スプライン関数などを用いてベクトル化することができる。

【0062】

また、対象が内輪郭を持つ場合、白画素の輪郭線追跡を行い、抽出した白画素輪郭の点列を用いて、同様に部分的直線あるいは曲線で近似する。

【0063】

以上のように、輪郭の区分線近似を用いれば、任意形状の図形のアウトラインをベクトル化することができる。元原稿がカラーの場合は、カラー画像から図形の色を抽出してベクトルデータとともに記録する。

【0064】

さらに、図12に示す様に、ある区間で外輪郭と、内輪郭あるいは別の外輪郭が近接している場合、2つの輪郭線をひとまとめにし、太さを持った線として表現することができる。具体的には、ある輪郭の各点 P_i から別輪郭上で最短距離となる点 Q_i まで線を引き、各距離 PQ_i が平均的に一定長以下の場合、注目区間は PQ_i 中点を点列として直線あるいは曲線で近似し、その太さは PQ_i の平均値とする。線や線の集合体である表罫線は、前記のような太さを持つ線の集合として効率よくベクトル表現することができる。

【0065】

尚、先に文字ブロックに対する文字認識処理を用いたベクトル化を説明したが、該文字認識処理の結果、辞書からの距離が最も近い文字を認識結果として用いるが、この距離が所定値以上の場合は、必ずしも本来の文字に一致せず、形状が類似する文字に誤認識している場合が多い。従って、本発明では、この様な文字に対しては、上記した様に、一般的な線画と同じに扱い、該文字をアウトライン化する。即ち、従来文字認識処理で誤認識を起こす文字に対しても誤った文字にベクトル化されず、可視的にイメージデータに忠実なアウトライン化によるベクトル化が行える。

【0066】

なお、写真と判定されたブロックに対しては本発明では、ベクトル化出来ない為、イメージデータのままとする。

【0067】

上述したように任意形状の図形のアウトラインをベクトル化した後、これらベクトル化された区分線を図形オブジェクト毎にグループ化する処理について説明する。

【0068】

図13は、ベクトルデータを図形オブジェクト毎にグループ化するまでのフロ

ーチャートを示している。まず、各ベクトルデータの始点、終点を算出する（7 0 0）。次に各ベクトルの始点、終点情報を用いて、図形要素を検出する（7 0 1）。図形要素の検出とは、区分線が構成している閉図形を検出することである。検出に際しては、閉形状を構成する各ベクトルはその両端にそれぞれ連結するベクトルを有しているという原理を応用し、検出を行う。次に図形要素内に存在する他の図形要素、もしくは区分線をグループ化し、一つの図形オブジェクトとする（7 0 2）。また、図形要素内に他の図形要素、区分線が存在しない場合は図形要素を図形オブジェクトとする。

【0 0 6 9】

図 1 4 は、図形要素を検出するフローチャートを示している。先ず、ベクトルデータより両端に連結していない不要なベクトルを除去し、閉図形構成ベクトルを抽出する（7 1 0）。次に閉図形構成ベクトルの中から該ベクトルの始点を開始点とし、時計回りに順にベクトルを追っていく。開始点に戻るまで行い、通過したベクトルを全て一つの図形要素を構成する閉図形としてグループ化する（7 1 1）。また、閉図形内部にある閉図形構成ベクトルも全てグループ化する。さらにまだグループ化されていないベクトルの始点を開始点とし、同様の処理を繰り返す。最後に、7 1 0 で除去された不要ベクトルのうち、7 1 1 で閉図形としてグループ化されたベクトルに接合しているものを検出し一つの図形要素としてグループ化する（7 1 2）。

【0 0 7 0】

以上によって図形ブロックを個別に再利用可能な個別の図形オブジェクトとして扱う事が可能になる。

【0 0 7 1】

<<アプリデータへの変換処理>>

ところで、一頁分のイメージデータをブロック毎にベクトル化処理（ステップ 1 2 9）した結果は、図 1 5 に示す様な中間データ形式のファイルとして変換されている。このようなデータ形式はドキュメント・アナリシス・アウトプット・フォーマット（D A O F）と呼ばれる。

【0 0 7 2】

図 1 5 は D A O F のデータ構造を示す図である。図 1 5 において、7 9 1 は H e a d e r であり、処理対象の文書画像データに関する情報が保持される。レイアウト記述データ部 7 9 2 では、文書画像データ中の T E X T (文字)、T I T L E (タイトル)、C A P T I O N (キャプション)、L I N E A R T (線画)、P I C T U R E (自然画)、F R A M E (枠)、T A B L E (表)等の属性毎に認識された各ブロックの属性情報とその矩形アドレス情報を保持する。文字認識記述データ部 7 9 3 では、T E X T、T I T L E、C A P T I O N等の T E X T ブロックを文字認識して得られる文字認識結果を保持する。表記述データ部 7 9 4 では、T A B L E ブロックの構造の詳細を格納する。画像記述データ部 7 9 5 は、P I C T U R E や L I N E A R T 等のブロックのイメージデータを文書画像データから切り出して保持する。

【 0 0 7 3 】

このような D A O F は、中間データとしてのみならず、それ自体がファイル化されて保存される場合もあるが、このファイルの状態では、所謂一般の文書作成アプリケーションで個々のオブジェクトを再利用する事は出来ない。そこで、次に、図 1 6 を用いて、この D A O F からアプリデータに変換する処理（ステップ 1 3 0）について詳説する。

【 0 0 7 4 】

ステップ 8 0 0 0 では、D A O F データの入力を行う。

【 0 0 7 5 】

ステップ 8 0 0 2 は、アプリデータの元となる文書構造ツリー生成を行う。

【 0 0 7 6 】

ステップ 8 0 0 4 は、文書構造ツリーを元に、D A O F 内の実データを流し込み、実際のアプリデータを生成する。

【 0 0 7 7 】

図 1 7 は、ステップ 8 0 0 2 での文書構造ツリー生成の詳細フロー、図 1 8 は、文書構造ツリーの説明図である。全体制御の基本ルールとして、処理の流れはマイクロブロック（単一ブロック）からマクロブロック（ブロックの集合体）へ移行する。以後ブロックとは、マイクロブロック、及びマクロブロックについて示す

ものとする。

【0078】

ステップ8100では、ブロック単位で縦方向の関連性を元に再グループ化する。スタート直後はマイクロブロック単位での判定となる。ここで、関連性とは、距離の近さや、ブロック幅（横方向の場合は高さ）がほぼ同一であることなどで定義することができる。また、距離、幅、高さなどの情報はD A O Fを参照し、抽出する。

【0079】

図18（a）はページ構成例、（b）はその文書構造ツリーである。ステップ8100の結果、T3、T4、T5が同じ階層のグループV1として生成され、T6、T7が同じ階層のグループV2としてまず生成される。

【0080】

ステップ8102では、縦方向のセパレータの有無をチェックする。セパレータは、例えば物理的にはD A O F中でライン属性を持つオブジェクトである。また論理的な意味としては、アプリ中で明示的にブロックを分割する要素である。ここでセパレータを検出した場合は、同じ階層で再分割する。

【0081】

ステップ8104では、分割がこれ以上存在し得ないか否かをグループ長を利用して判定する。ここで、縦方向のグループ長がページ高さ（該ページに存在する複数のブロックの最上端と際下端の距離）となっている場合は、文書構造ツリー生成は終了する。図18の場合は、グループV1、V2には、セパレータもなく、グループ高さはページ高さではないので、ステップ8106に進む。

【0082】

ステップ8106では、ブロック単位で横方向の関連性を元に再グループ化する。ここもスタート直後の第一回目はマイクロブロック単位で判定を行うことになる。このときの関連性、及びその判定情報の定義は、縦方向の場合と同じである。

【0083】

図18の場合は、T1、T2でグループH1が生成され、また、V1、V2で

グループ H 2 が生成される。グループ H 1 とグループ H 2 は、V 1, V 2 の 1 つ上の同じ階層のグループとして生成される。

【 0 0 8 4 】

ステップ 8 1 0 8 では、横方向セパレータの有無をチェックする。図 1 8 では、S 1 があるので、これをツリーに登録し、H 1、S 1、H 2 という階層が生成される。

【 0 0 8 5 】

ステップ 8 1 1 0 は、分割がこれ以上存在し得ないか否かをグループ長を利用して判定する。ここで、横方向のグループ長がページ幅となっている場合は、文書構造ツリー生成は終了する。

【 0 0 8 6 】

そうでない場合は、8 1 0 2 に戻り、再びもう一段上の階層で、縦方向の関連性チェックから繰り返す。

【 0 0 8 7 】

図 1 8 の場合は、分割幅がページ幅（該ページに存在する複数のブロックの最左端と最右端の距離）になっているので、ここで終了し、最後にページ全体を表す最上位階層の V 0 が文書構造ツリーに付加される。

【 0 0 8 8 】

文書構造ツリーが完成した後、その情報を元に 8 0 0 4 においてアプリデータの生成を行う。図 1 8 の場合は、具体的には、以下のようになる。

【 0 0 8 9 】

すなわち、H 1 は横方向に 2 つのブロック T 1 と T 2 があるので、2 カラムとし、T 1 の内部情報（D A O F を参照、文字認識結果の文章、画像など）を出力後、カラムを変え、T 2 の内部情報出力、その後 S 1 を出力となる。

【 0 0 9 0 】

H 2 は横方向に 2 つのブロック V 1 と V 2 があるので、2 カラムとして出力、V 1 は T 3、T 4、T 5 の順にその内部情報を出力、その後カラムを変え、V 2 の T 6、T 7 の内部情報を出力する。

【 0 0 9 1 】

以上により出力された順番でアプリデータへの変換処理が行えるので、例えば文字領域の読み順などを正しい順番でアプリデータへ変換できる。

【0092】

<<ポータルシート作成>>

まず、図19を参照してポータルシート2201について説明する。図19のポータルシートには、6つの原稿の情報（ステップ120で入力されたイメージに係る情報）が記録されている。それぞれは、以下の2202-2205の情報からなっている。

【0093】

2203は、原稿の縮小画像（サムネール）で、1ページ目を表している。

【0094】

2202, 2204は日付、タイトル、作者、ID、キーワード、要約等であり、サムネール2203に関連する情報で、予め原稿に付随している情報であったり、ユーザが入力した情報であったり、原稿の内容から自動的に抽出された情報であったりする。

【0095】

2205は原稿の蓄積場所等を示すためのポインタである。ここでは二次元バーコードで表されているが、1次元バーコードであったり、電子透かしであっても問わない。

【0096】

次に、図3の134の処理にあたる、ポータルシートの作成方法について、図20を使用して説明する。

【0097】

まず操作者が、あらかじめ操作パネル（図示せず）よりポータルシートを出力するモードを選択し、1枚に表示する件数（ページ内原稿数）を操作パネルのテンキーにて入力する。本例では、6が設定されているものとする。

【0098】

まず、ステップ2211において、ポータルシートのページ内に原稿が配置されているかを確認する。最初は、原稿が配置されていないので、2212にてポ

ータルシートページの初期化処理を行なう。

【0099】

次に前述した各原稿の日付やタイトル等の原稿情報を取得し（2213）、それらを図19のようにポータルシートページ上に配置する（2214）。また、ステップ2214では、各原稿のポインタ情報も付加する。

【0100】

次にポータルシートページ内に配置した原稿数が、操作者が指定したページ内原稿数に達しているかどうかを確認し（2215）、もし達していた場合には現ページをプリンタへ出力する。

【0101】

最後にすべての原稿を配置したかどうかを確認し（2217）、まだ未配置の原稿が存在すれば2211の処理に戻り、すべて配置し終わっていたら、必要に応じて最後のページを出力して（2218, 2219）、終了する。

【0102】

次に、ステップ2213で取得され2214で配置される原稿情報（タイトル、キーワード、ID、日付、著作者、要約、目次など）の作成方法について説明する。

【0103】

原稿に含まれるテキストデータに対してタイトルを抽出する方法としては、例えば、前述の方法により抽出したレイアウト情報およびフォント情報を利用し、テキストデータの最初の部分に現れるフォントの大きい文で、かつテキストデータ中に使用される頻度の高い文をタイトルとして抽出するようにする。

【0104】

原稿に含まれるテキストデータに対してキーワードを抽出する方法としては、例えば、まずテキストデータ全体を形態素解析などにより、単語に分解する。全単語を使用頻度によりソートし、頻度の高いものから順にキーワードとする。さらに、より有効なキーワードを抽出するには、予め用意したキーワード用のデータベースと照合するようにしても良い。

【0105】

I D、日付、著作者の情報に関しては、電子ファイルの検索によりファイルが存在した場合に、そのファイルの属性情報として取得する。

【0 1 0 6】

要約情報に関しては、文字コード列で構成されるテキストデータに対して、その要約を作成する方法としては、次のようなものがある。まずテキストデータを文あるいは文節単位に分解する。これは句読点の文字コードを境界として機械的に分割することができる。次にそれぞれの文あるいは文節に対し重要度を計算して付与する。この重要度の計算方法としては、テキストデータ全体に含まれる単語に対して出現頻度を求め、頻繁に出現する単語に高い点数を与え、各文あるいは文節の重要度は、その中に含まれる各単語の点数の総和とする方法、レイアウト情報やフォント情報を用いて、タイトルや見出しに相当する文を検出し、その文自体の重要度を高めたり、文に含まれる単語の点数を上げて重要度を計算したりする方法、などがある。最後に、重要度が上位の文あるいは文節を抽出することで要約文が作成される。このとき、上位何位まで抽出するかを変化させて、所望のサイズの要約文を生成することが可能である。

【0 1 0 7】

目次情報の作成に関しては、例えば、複数ページに渡るようなテキストデータに対して目次を作成する場合には、テキストをレイアウト上の空白などから段落に分割し、それぞれに要約を施して一行程度の要約文を割りあてればよい。特に、前述のようにレイアウト情報やフォント情報からタイトルや見出しとして抽出された文は、目次として最適である。

【0 1 0 8】

次に、図 1 9 の 2 2 0 5 等 に示されている原稿の蓄積場所を示すためのポイント情報付加処理について説明する。このポイント情報付加処理は、ステップ 2 2 1 4 の原稿情報配置時に行われるものとする。

【0 1 0 9】

図 2 1 はポインター情報としてのデータ文字列を 2 次元バーコード（QR コードシンボル：J I S X 0 5 1 0）3 1 1 にて符号化する過程を示すフローチャートである。

【0 1 1 0】

2次元バーコード内に組み込むデータは、対応するファイルのアドレス情報を表しており、例えばファイルサーバ名およびファイル名からなるパス情報で構成される。或いは、対応するファイルへのURLや、対応するファイルの格納されているデータベース105内あるいはMFP100自体が有する記憶装置内で管理されるファイルID等で構成される。

【0 1 1 1】

まず、符号化する種類の異なる文字を識別するため、入力データ列を分析する。また、誤り検出及び誤り訂正レベルを選択し、入力データが収容できる最小型番を選択する。(ステップ900)。

【0 1 1 2】

次に、入力データ列を所定のビット列に変換し、必要に応じてデータのモード(数字、英数字、8ビットバイト、漢字等)を表す指示子や、終端パターンを付加する。さらに所定のビットコード語に変換する。(ステップ901)。

【0 1 1 3】

この時、誤り訂正を行うため、コード語列を型番および誤り訂正レベルに応じて所定のブロック数に分割し、各ブロック毎に誤り訂正コード語を生成し、データコード語列の後に付加する(ステップ902)。

【0 1 1 4】

該ステップ902で得られた各ブロックのデータコード語を接続し、各ブロックの誤り訂正コード語、必要に応じて剰余コード語を後続する。(ステップ903)。

【0 1 1 5】

次に、マトリクスに位置検出パターン、分離パターン、タイミングパターンおよび位置合わせパターン等とともにコード語モジュールを配置する。(ステップ904)。

【0 1 1 6】

更に、シンボルの符号化領域に対して最適なマスクパターンを選択して、マスク処理パターンをステップ904で得られたモジュールにXOR演算により変換

する（ステップ 905）。

【0117】

最後に、ステップ 905 で得られたモジュールに形式情報および型番情報を生成して、2次元コードシンボルを完成する（ステップ 906）。

【0118】

上記に説明した、アドレス情報の組み込まれた2次元バーコードは、ステップ 2214 の原稿情報配置時に行われるものとするが、例えば、クライアント PC 102 からポータルシートを生成する場合に、データ処理装置 115 内で記録可能なラスターデータに変換された後にラスターデータ上の所定の個所に付加されて画像形成されるようにしても構わない。

【0119】

尚、同様の目的で付加情報を付与する手段は、本実施例で説明した2次元バーコードの他に、例えば、ポインター情報を直接文字列で文書に付加する方法、文書内の文字列、特に文字と文字の間隔を微小に変調して情報を埋め込む方法、文書中の中間調画像中に埋め込む方法等、一般に電子透かしと呼ばれる方法が適用出来る。

【0120】

以下電子透かしを適用して、ポータルシート上に掲載される中間調画像（サムネール画像）中へコード情報を埋め込む例について説明する。

【0121】

コード情報はバイナリデータ列として画像上の各色成分一画素について一ビットを重み付けて埋め込まれる。画素にコード情報を埋め込む際、近傍画素で構成される一つのブロックに対し一つのコード情報を埋め込む。コード情報は互いに重ならない複数のブロックに対して繰り返し埋め込まれる。ここで、同じコード情報が埋め込まれた複数のブロックが構成するブロックをマクロブロックとよぶ。なお、サムネール画像全体をマクロブロックとすることも可能である。埋め込まれたコード情報を取り出す際はマクロブロックよりブロック毎の相互相関を求め、その統計量よりコード情報が抽出できる。

【0122】

このような画像へのコード情報の埋め込みは人間の視覚では認識できないレベルで画素を変化させることで十分埋め込み可能であるが、ポータルシートで使用するサムネイル画像では、その目的上、画質はそれほど気にされないため、より重みを強くして埋め込むことで耐性を持った埋め込みも可能である。

【0 1 2 3】

＜＜ポータルシートの編集＞＞

次に、上述したようにして作成されたポータルシートについて説明する。

【0 1 2 4】

まず、スキャンしたポータルシートを使用して、指定ページ、指定段落、指定画像、指定文書の指定ページ、指定文書の指定ページの指定段落、指定文書の指定ページの指定画像等にアクセスするための所望のポータルシートの編集について、図 2 2、図 2 3 を使用して説明を行う。但し、ここで示す実施例は基本形であり、拡張は任意とする。なお、ここではポータルシートをスキャンして得たポータルシートの編集について述べるが、上述したようにして作成されたポータルシートをそのまま電子データとして保持しておき、その保持しておいたポータルシートを編集するようにしてもよい。

【0 1 2 5】

図 2 2 の 2 5 0 1 は、MFP などのユーザーインターフェース（以下 U I とする）の概略図である。前述したように、ポータルシートをスキャンすることにより、電子透かしなどから、サムネイル、タイトル、日付、要約、キーワード、ポインター情報などの情報を得られるため、これらを U I 上に表示し、ユーザは、その情報を見て所望とする新規ポータルシートの編集を行う。

【0 1 2 6】

以下、U I に表示されている各図について説明を行う。

【0 1 2 7】

2 5 0 2 ～ 2 5 0 6 は、表示されているサムネイルを利用してポータルシートを編集するための、ボタンであり、それぞれ順に、“追加”、“削除”、“全選択”、“部分情報へアクセス”、“終了”、などの機能を備える。また、2 5 3 0、2 5 3 1 は、複数のポータルシートをスキャンした場合のため、“前のペー

ジへ戻る”、“次のページへ進む”の機能を備えるボタンである。2 5 1 0～2 5 1 5 は、ポータルシートのスキャン後に得られるサムネイル及びタイトル、日付などの情報を示す図である。2 5 2 0～2 5 2 5 は、編集後のポータルシート、（以下、新規ポータルシート）を示す図である。

【0 1 2 8】

図 2 2 におけるポータルシートの編集操作について説明する。ユーザは、画面左（スキャンしたポータルシート画面）の所望とするサムネイルに、指またはペンで触れ、サムネイルをアクティブな状態にした後、追加ボタン 2 5 0 2 を押すことにより、画面右（新規ポータルシート画面）にそのサムネイルを追加することが可能である。また、全選択ボタン 2 5 0 4 を押すことにより、画面左に表示されている全てのサムネイルをアクティブな状態にすることも可能である。新規ポータルシートから削除したいサムネイルは、削除ボタン 2 5 0 3 を押すことにより削除可能である。次のスキャンしたポータルシートを画面左に表示させたい場合は、ページ変更ボタン 2 5 3 0，2 5 3 1 を押す。そして、所望の編集終了後、“編集の終了”ボタン 2 5 0 6 を押すことで作業を終了する。図 2 2 では、2 5 1 1、2 5 1 5 を選択（アクティブな状態）し、追加ボタンを押した結果を示している。なお、ボタンによる操作以外にも、ドラッグ&ドロップでポータルシートの編集を行うことも可能である。具体的には、スキャンしたポータルシート画面の所望とするサムネイルに触れたままドラッグして、新規ポータルシート画面の所望の追加位置にドロップする（離す）ことにより、追加することができる。なお、ドロップした追加位置に既にサムネイルが登録されていた場合は、既に登録されていたサムネイルを順にずらして、該追加位置にドロップしたサムネイルを追加する。また、新規ポータルシート画面の所望のポータルシートに指で触れ、そのままドラッグして新規ポータルシート画面外の位置にドロップすることにより、新規ポータルシートから該ポータルシートを削除することができる。また、ドラッグ&ドロップにより、新規ポータルシート画面に登録されているサムネイルの順序を変更することも可能である。例えば新規ポータルシート画面において、サムネイル F をドラッグして 2 5 2 0 の位置でドロップすると、サムネイル B の登録順がずれて 2 5 2 1 の位置に表示されるとともにサムネイル F は 2

5 2 0 の位置に表示されることになる。

【0 1 2 9】

また、不図示ではあるが、タイトルその他の情報を編集することも可能とする。また、これらの画面右の編集結果は、画面左のポータルシートの内部的な情報（電子透かしによるオリジナル画像へのポインタ情報など）を継承し、新規ポータルシートとして作成される。このようにして、必要な情報のみを集めた新規ポータルシートを作成することが可能になる。

【0 1 3 0】

図 2 3 は図 2 2 にある“部分情報へアクセス”（2 5 0 5）を選択した際に表示される U I 画面である。画面左側の 2 6 3 0 は、図 2 2 のポータルシート中から選択したサムネイルのポインター情報に基づいてオリジナル電子データ（原稿ページ単位のデータ）を表示するものである。このオリジナル電子データは、タイトルや本文やイラストなどの属性で領域分割され、この分割された領域毎のオブジェクトが識別可能な状態で表示されている（2 6 3 1，2 6 3 2）。図 2 3 は、このオブジェクトを、図 2 2 のポータルシートに追加することを可能とするための U I 画面を提供するものである。

【0 1 3 1】

2 6 3 3 は現在表示しているサムネイルのページ番号を表示するもので、2 6 3 4 と 2 6 3 5 を用いて他のページに移動することができる。2 6 4 3 ～2 6 4 6 は、左側に表示されたオリジナル電子データから選択されたオブジェクトを表示するもので、2 6 4 0 ～2 6 4 2 により表示を移動させることが可能となる。また、オブジェクトの追加・削除の機能、および編集終了の機能は、ボタン 2 6 5 0 ～2 6 5 2 を用いて行われる。なお、ドラッグ&ドロップによるオブジェクトの追加・削除・順序変更などの編集操作を図 2 2 と同様に行うことも可能である。

【0 1 3 2】

以下、ポータルシートをスキャンして得られた情報を図 2 2 のようにサムネイルとして U I 上に表示して、そのサムネイルとして表示されている文書の部分情報（指定ページ、指定段落、指定画像等）を使用して新規ポータルシートを作成

する際の詳細な説明を行う。ユーザはポータルシートをスキャンして得られたサムネイル表示（2 5 1 0 ～ 2 5 1 5）のうち、部分情報にアクセスしたいものを選択し、“部分情報にアクセス”（2 5 0 5）を選択する。すると、図 2 3 に示されているような U I 画面が表示され、選択したサムネイルのオリジナル電子データが 2 6 3 0 として表示される。さらに、この 2 6 3 0 はベクトル化されたオブジェクト（段落、画像等）毎に分割されているのが一目でわかるようになっており、ユーザがアクセス（使用）したい部分情報にタッチ選択し、“追加” ボタンを押下することにより、選択対象の部分情報（オブジェクト）を示す右側の画面に、追加することができる。また、指定ページにアクセスしたい場合は 2 6 3 4、2 6 3 5 でページを移動させることによりアクセスが可能となる。また、追加された部分情報は“削除” ボタンにより削除を行うことができる。また、“終了” ボタンにより、部分情報へのアクセスは終了して図 2 2 に戻り、図 2 3 で選択した部分情報（図 2 3 の右側画面に表示されているオブジェクト）は、順に図 2 2 の新規ポータルシートに追加（2 5 2 0 ～ 2 5 2 5）されることになる。このようにして、“部分情報にアクセス” ボタンを用いることにより、部分情報（指定ページ、指定段落、指定画像等）単位で新規ポータルシートに追加することも可能になり、必要な部分だけをポータルシートにすることができる。

【0 1 3 3】

<<ポータルシートの利用>>

上述したように作成あるいは編集作成されたポータルシートを用いて、閲覧や送信や印刷などの処理を行いたいページやオブジェクトを選択することが容易になる。例えば、ポータルシートをスキャンすると、図 2 2 左側のように、スキャンしたポータルシートが表示され、該ポータルシートから処理を行いたいページ（あるいはオブジェクト）を選択し、所望の処理モードを選択して“S T A R T” ボタンを押下すると、該選択されたページ（あるいはオブジェクト）のポインター情報からオリジナル電子データを取得し、該オリジナル電子データに対して所望の処理（閲覧や送信や印刷など）が実行される。

【0 1 3 4】

（その他の実施形態）

＜ファイルアクセス権に関する別実施例＞

我々が扱う文書ファイルの中には、第三者による再利用を制限すべき物がある。先の実施例ではファイルサーバに蓄積されたファイルは全て自由にアクセス出来、ファイル全体、或いはその一部のオブジェクトは全て再利用が可能な事を前提に説明した。そこで、先の実施例でポインター情報からファイルを検索した際に、検索の結果、特定出来たファイルにアクセス権の制限が有る場合についての別実施例を図9を用いて説明する。ステップ403までは先の実施例と同様の為、説明は省略する。ステップ403でファイルが特定された場合、ステップ404でファイルサーバはそのファイルのアクセス権情報を調べ、アクセス制限がある場合には、MFPに対してパスワードの送信を要求する（ステップ405）。

【0135】

MFPは操作者に対してパスワードの入力を促し、入力されたパスワードをファイルサーバに送信する（ステップ406）。

【0136】

ファイルサーバは送信されたパスワードを照合して認証を行い（ステップ407）、認証成功した場合には、図3で説明した様に、ファイルのアドレスを通知（ステップ134）すると共に、ユーザの希望する処理が画像ファイルデータの取得であれば、MFPに対してファイルを転送する（ステップ408）。

【0137】

尚、アクセス権の制御を行う為の認証の方法は、ステップ405、406に示したパスワードによる方法に限定されず、例えば、指紋認証等の一般に広く用いられている生体認証、カードによる認証等、様々な認証手段を用いる事が出来る。

【0138】

又、本別実施例では紙文書に付加的に付与されたポインター情報によりファイルを特定した場合の実施例を示したが、図3のステップ126-128で示す所謂検索処理でファイルを特定した場合においても同様の制御が可能である。

【0139】

一方、図3のステップ129で説明したベクトル化処理に対しても、制限を加

える事が出来る。即ち紙文書を走査して得られたイメージ情報から、透かし等により該文書に対してのアクセス権の制限の存在を検出した場合には、認証確認が取れた場合のみベクトル化処理を行う事で、機密性の高い文書の使用に制限をかける事が出来る。

【0 1 4 0】

＜ファイル特定に関する別実施例＞

先の実施例では原稿走査して得られるイメージ情報から元ファイルデータを特定する手段は、図 3 に示す様に、文書中に付与されたポインター情報に基づいて特定するか、或いは文書中に記載された各オブジェクト情報に基づいて対応する電子ファイルを検索するかのいずれかに依るが、より正確に元ファイルを特定するには、該両手段を満足する電子ファイルを特定すれば良い。即ち、原稿中から得られるポインター情報から元ファイルの存在が検出出来たとしても、更に該文書中のオブジェクト情報を使って、例えば、レイアウト情報に従うレイアウト検索、文字認識されたキーワードによる全文検索などを検出されたファイルに対して行い、高い一致が得られた場合に、該検出したファイルを、正式に元ファイルであると特定する。これは、例えば、ポインター情報の下位の部分が曖昧で有ったり、誤り訂正でも訂正できなかった場合に対して、検索の範囲を絞り込んでファイルを特定出来る為、より高速で、確度の高いファイル特定が行える。

【0 1 4 1】

＜ベクトル化の別実施例＞

先の実施例では検索手段で、元ファイルの特定が出来ない場合、イメージ画像全体に対してベクトル化処理を行うが、例えば、一般の文書の場合、文書中のオブジェクト全て新規に作成された物でなく、一部のオブジェクトは他のファイルから流用して作成される場合がある。例えば、背景オブジェクト（壁紙）は文書作成アプリケーションで幾つかのパターンを予め用意してあり、その中から選択して用いるのが通常である。従って、このようなオブジェクトは文書ファイルデータベースの中の他の文書ファイル中に存在している可能性が高く、又再利用可能なベクトルデータとして存在する可能性が高い。

【0 1 4 2】

そこで、図3のベクトル化処理129の別実施例として、ブロックセレクション処理で個別のオブジェクトに分割された各オブジェクトに対して、該オブジェクト単位でデータベース中から該一致するオブジェクトを一部に含むファイルを検索し、一致したオブジェクトに対して個別に該ファイルからオブジェクト単位でベクトルデータを取得する。これに依って、文書全体をベクトル化する必要が無くなり、より高速にベクトル化出来、更にベクトル化による画質劣化を防止出来る。

【0143】

一方、図3において検索処理126-128で元ファイルがPDFとして特定できた場合、該PDFがその文書の文字オブジェクトに対して既に文字認識された文字コードを付加ファイルとして有している場合がある。このようなPDFファイルをベクトル化する際には、該文字コードファイルを用いれば、129以降のベクトル化処理の中の文字認識処理を省く事が出来る。即ち、ベクトル化処理をより高速に処理する事が可能に成る。

【0144】

<ポータルシート化対象の別実施例>

先の実施例では制御対象文書は、文書イメージのページ単位でポータルシート化していたが、何もこれに限ることは無く、複数文書のページ単位でポータルシートを生成しても良いし、文書単位でその表紙等をサムネイルとして表示してポータルシートを作成するようにしても良い。また、文書単位、ページ単位の情報混在していても良い。

【0145】

上述した実施形態1及びその他の実施形態の記載に基づき、本発明を以下に総括する。

【0146】

(実施態様1)

入力されたイメージそれぞれに対して、格納手段に格納されているオリジナル電子データを検索する検索ステップと、前記検索ステップでオリジナル電子データが検索されなかった場合、前記入力されたイメージをベクトルデータ化するべ

クトル化ステップと、前記ベクトル化ステップでベクトルデータ化されたイメージを電子データとして格納手段に格納する格納ステップと、前記検索ステップでオリジナル電子データが検索された場合において当該検索されたオリジナル電子データに関する情報あるいは前記検索ステップでオリジナル電子データが検索されなかった場合において前記ベクトル化ステップでベクトル化されて前記格納ステップで格納された電子データに関する情報の少なくともいずれかを含むシートを生成するシート生成ステップとを有することを特徴とする画像処理方法。

【0 1 4 7】**(実施態様 2)**

前記シート生成ステップで生成されたシートに含まれる、オリジナル電子データに関する情報あるいは前記格納ステップで格納された電子データに関する情報は、それぞれ、当該検索されたオリジナル電子データの格納アドレスあるいは前記格納ステップで格納された電子データの格納アドレスを含むことを特徴とする請求項 1 に記載の画像処理方法。

【0 1 4 8】**(実施態様 3)**

前記検索ステップでは、前記入力されたイメージに付加されている、前記オリジナル電子データが格納されているアドレスを示す識別子を認識し、該識別子を認識した結果に基づいて前記オリジナル電子データを検索することを特徴とする実施態様 1 に記載の画像処理方法。

【0 1 4 9】**(実施態様 4)**

前記ベクトル化ステップでは、前記イメージ内に含まれる文字ブロックを文字認識した結果に基づいてベクトル化することを特徴とする実施態様 1 に記載の画像処理方法。

【0 1 5 0】**(実施態様 5)**

前記ベクトル化ステップでは、有意な画素の輪郭に基づいてアウトライン化してベクトル化することを特徴とする実施態様 1 に記載の画像処理方法。

【0151】**(実施態様6)**

更に、前記ベクトル化ステップでベクトルデータ化されたイメージを、既存の文書作成アプリケーションで扱える既定のフォーマットに変換するフォーマット変換ステップを有することを特徴とする実施態様1に記載の画像処理方法。

【0152】**(実施態様7)**

前記ベクトル化ステップでは、前記イメージ中に含まれるオブジェクトごとにベクトルデータ化することを特徴とする実施態様1に記載の画像処理方法。

【0153】**(実施態様8)**

前記シート生成ステップで生成されたシートに含まれる、オリジナル電子データに関する情報あるいは前記格納ステップで格納された電子データに関する情報は、それぞれの電子データのサムネイル、タイトル、要約、キーワード、ID、日付、著作者の少なくともいずれかを含むことを特徴とする実施態様1に記載の画像処理方法。

【0154】**(実施態様9)**

更に、前記生成されたシートを編集する編集ステップを有することを特徴とする実施態様1に記載の画像処理方法。

【0155】**(実施態様10)**

前記編集ステップでは、前記シートに含まれるオリジナル電子データに関する情報あるいは前記格納ステップで格納された電子データに関する情報の中から、所望のデータを選択して新規シートを編集して作成することを特徴とする実施態様9に記載の画像処理方法。

【0156】**(実施態様11)**

前記編集ステップでは、前記シートに含まれるオリジナル電子データに関する

情報あるいは前記格納ステップで格納された電子データに関する情報の中から、所望のデータを選択し、更に、当該選択したデータ中の所望の部分情報を選択することにより、当該選択された部分情報を用いて編集することを特徴とする実施態様 9 に記載の画像処理方法。

【0 1 5 7】**(実施態様 1 2)**

更に、前記生成されたシートをスキャンして得たイメージを表示し、当該表示されたイメージにおいてユーザにより選択された電子データに関する情報に基づいて、処理を実行する処理ステップを有することを特徴とする実施態様 1 に記載の画像処理方法。

【0 1 5 8】**(実施態様 1 3)**

前記処理ステップでは、前記ユーザにより選択された情報に対応するオリジナル電子データあるいは電子データに対して、所望の処理が実行されることを特徴とする実施態様 1 2 に記載の画像処理方法。

【0 1 5 9】**(実施態様 1 4)**

前記格納アドレスに関する情報は、電子透かしあるいはバーコードの少なくともいずれかで記載されることを特徴とする実施態様 2 に記載の画像処理方法。

【0 1 6 0】**(実施態様 1 5)**

実施態様 1 乃至 1 4 のいずれかに記載の各ステップを、コンピュータに実行させるためのコンピュータプログラム、あるいは該コンピュータプログラムを格納した記憶媒体も本発明を実現しうる。

【0 1 6 1】**(実施態様 1 6)**

入力されたイメージそれぞれに対して、格納手段に格納されているオリジナル電子データを検索する検索手段と、前記検索手段でオリジナル電子データが検索されなかった場合、前記入力されたイメージをベクトルデータ化するベクトル化

手段と、前記ベクトル化手段でベクトルデータ化されたイメージを電子データとして格納手段に格納する保存手段と、前記検索手段でオリジナル電子データが検索された場合において当該検索されたオリジナル電子データに関する情報あるいは前記検索手段でオリジナル電子データが検索されなかった場合において前記ベクトル化手段でベクトル化されて保存手段で格納された電子データに関する情報の少なくともいずれかを含むシートを生成するシート生成手段とを有することを特徴とする画像処理システム。

【0162】

【発明の効果】

以上説明したように、本発明によれば、入力されたイメージのポータルシートを作成するので、該ポータルシートを用いて、記憶手段に格納されている電子データを容易に再利用できるという効果がある。

【0163】

また、オリジナル電子データを特定するためのポインター情報が付加されているので、オリジナル電子データの検索も容易であるという効果がある。

【0164】

また、オブジェクト単位（ブロック単位）でポータルシートを編集作成できるので、必要な部分だけを扱うことができ、閲覧などの処理時の選択が容易になるという効果がある。

【0165】

また、ベクトルデータで格納しているので、記憶容量が少なくて済むという効果がある。また、オリジナル電子データが見つかった場合は、そのオリジナル電子データを用いるようにするので、画質の劣化も少なくて済むという効果がある。

【図面の簡単な説明】

【図1】

画像処理システム構成例

【図2】

MFP構成例

【図 3】

画像処理の全体動作の概要を示すフローチャート

【図 4】

ブロックセレクション処理の例

【図 5】

ブロック情報および入力ファイル情報の構成例

【図 6】

2 次元バーコードの復号処理フローチャート

【図 7】

2 次元バーコードが付加された原稿の例

【図 8】

ポインター情報によるファイル検索フローチャート

【図 9】

アクセス制限付きのファイル検索フローチャート

【図 1 0】

ファイルのレイアウト検索処理フローチャート

【図 1 1】

ベクトル化処理の説明図

【図 1 2】

ベクトル化処理の説明図

【図 1 3】

ベクトルデータのグループ化処理フローチャート

【図 1 4】

図形要素検出フローチャート

【図 1 5】

D A O F のデータ構造

【図 1 6】

アプリデータ変換フローチャート

【図 1 7】

文書構造ツリー作成フローチャート

【図 1 8】

文書構造ツリー作成説明図

【図 1 9】

ポータルシート of 例

【図 2 0】

ポータルシート作成処理フローチャート

【図 2 1】

ポインター情報符号化処理フローチャート

【図 2 2】

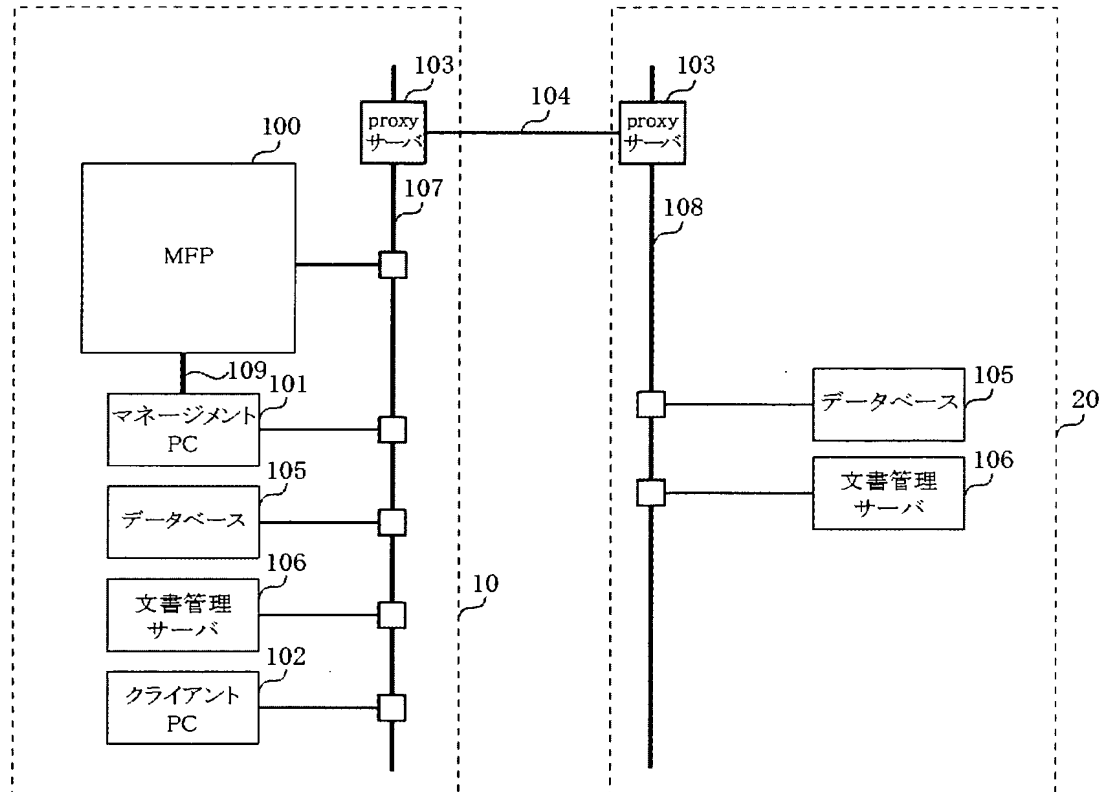
ポータルシート編集画面

【図 2 3】

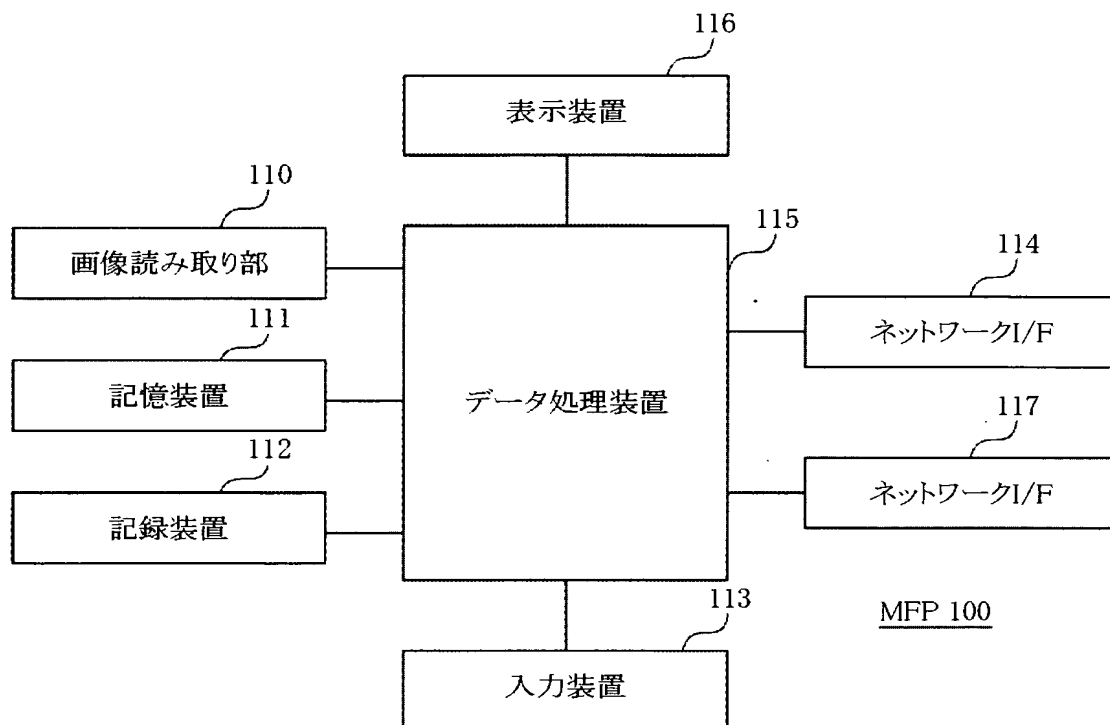
部分情報へのアクセス時のポータルシート編集画面

【書類名】 図面

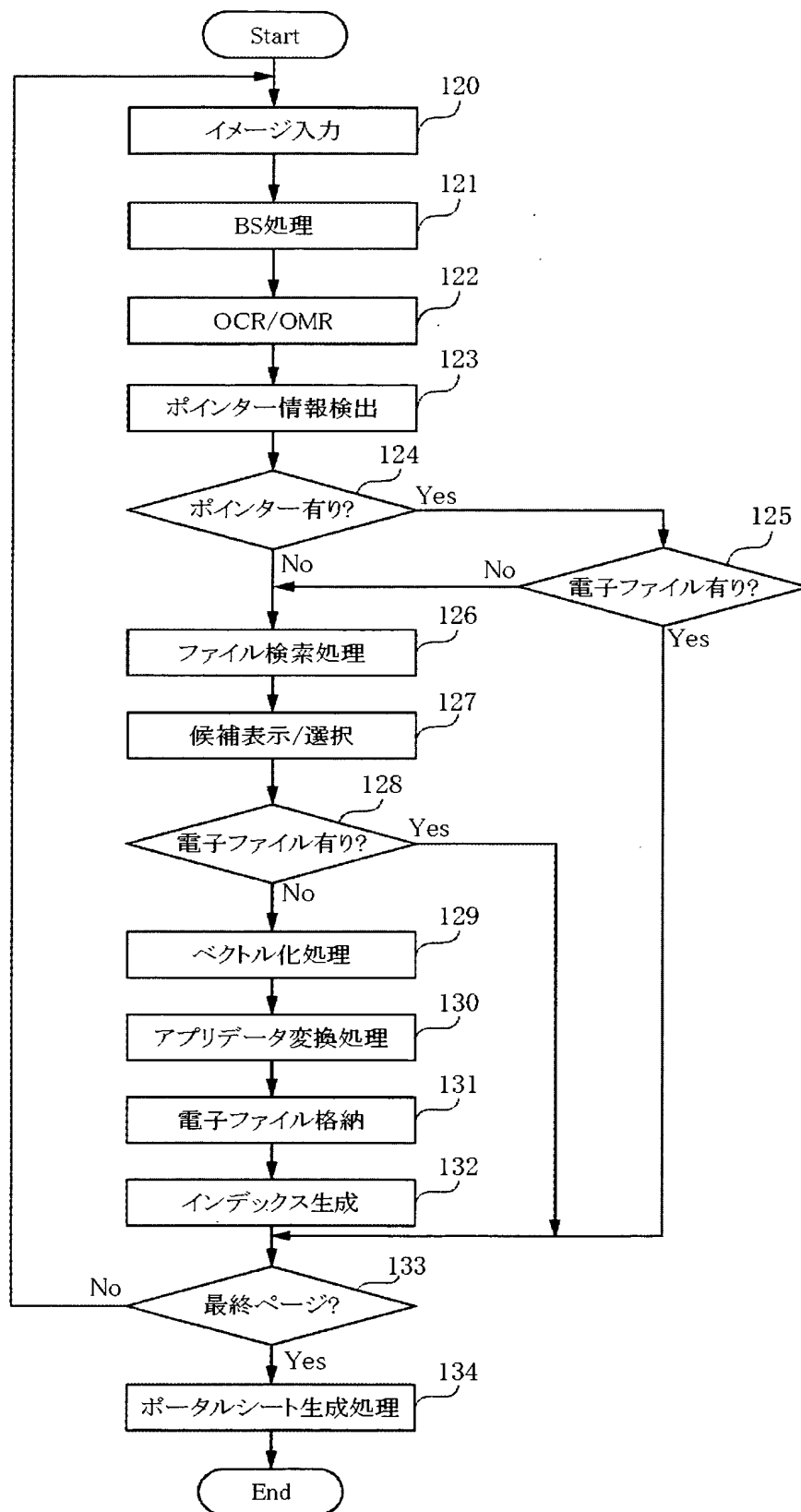
【図 1】



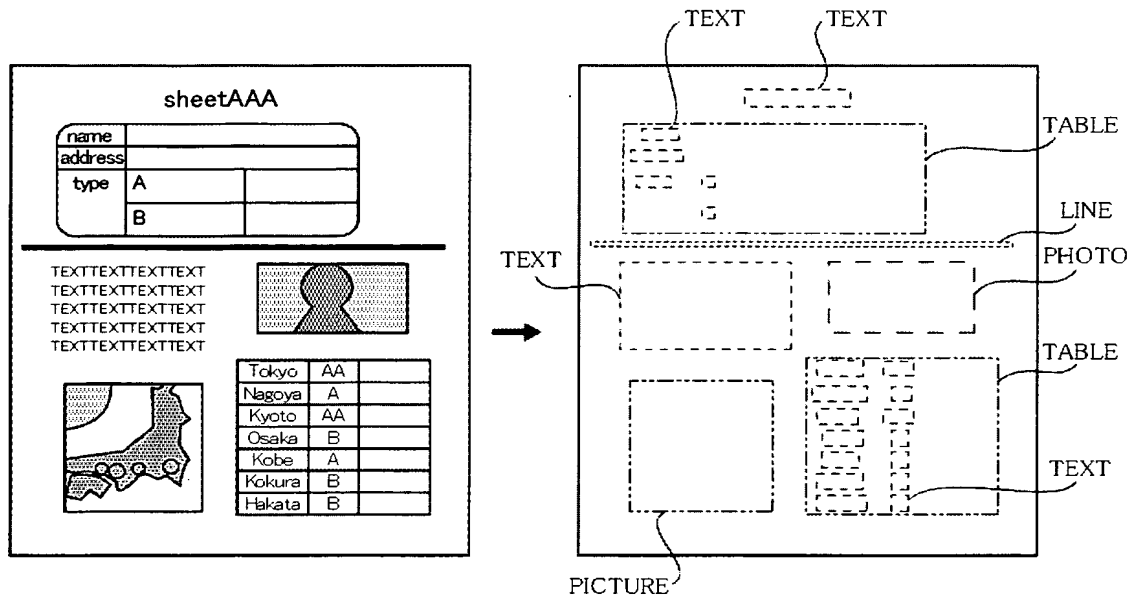
【図 2】



【図 3】



【図 4】



【図 5】

ブロック情報

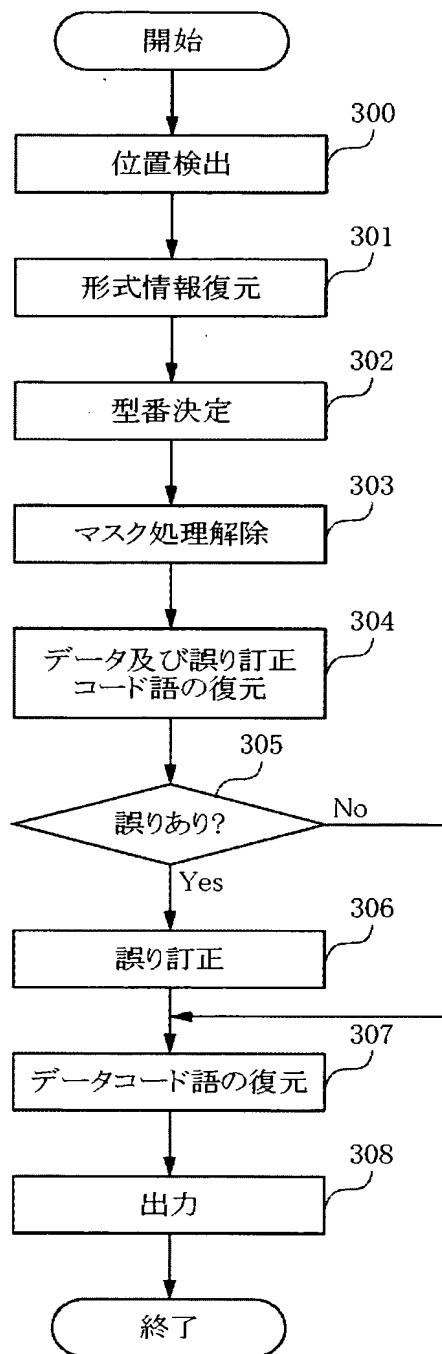
	属性	座標X	座標Y	幅W	高さH	OCR情報
ブロック1	1	X1	Y1	W1	H1	有
ブロック2	3	X2	Y2	W2	H2	有
ブロック3	2	X3	Y3	W3	H3	無
ブロック4	1	X4	Y4	W4	H4	有
ブロック5	3	X5	Y5	W5	H5	有
ブロック6	5	X6	Y6	W6	H6	無

* 属性 1:text 2:picture 3:table 4;line 5:photo

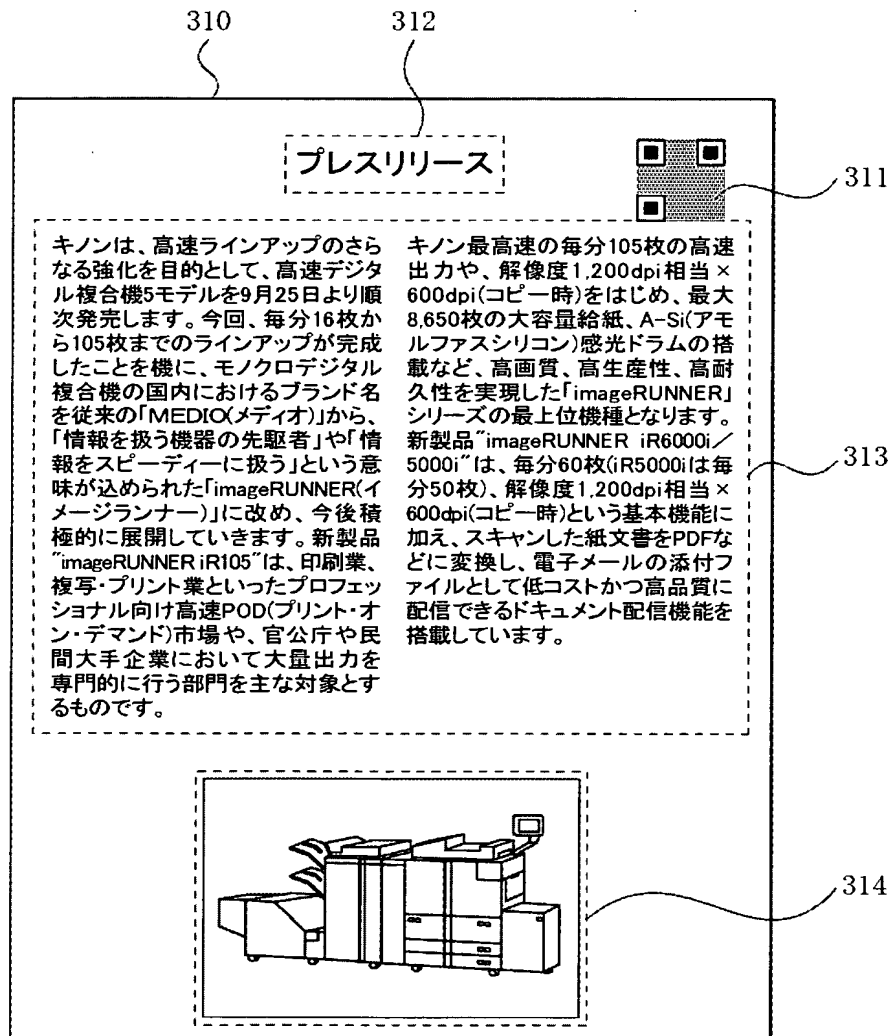
入力ファイル情報

ブロック総数	N(=6)
--------	-------

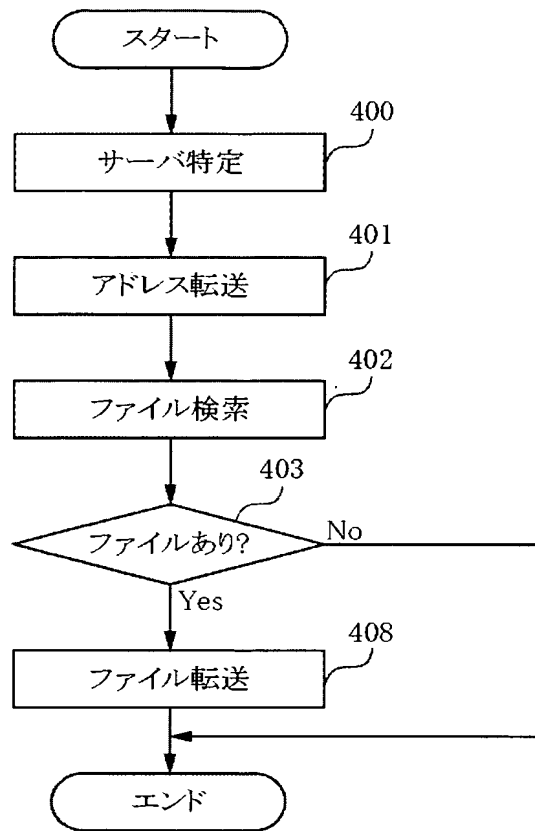
【図 6】



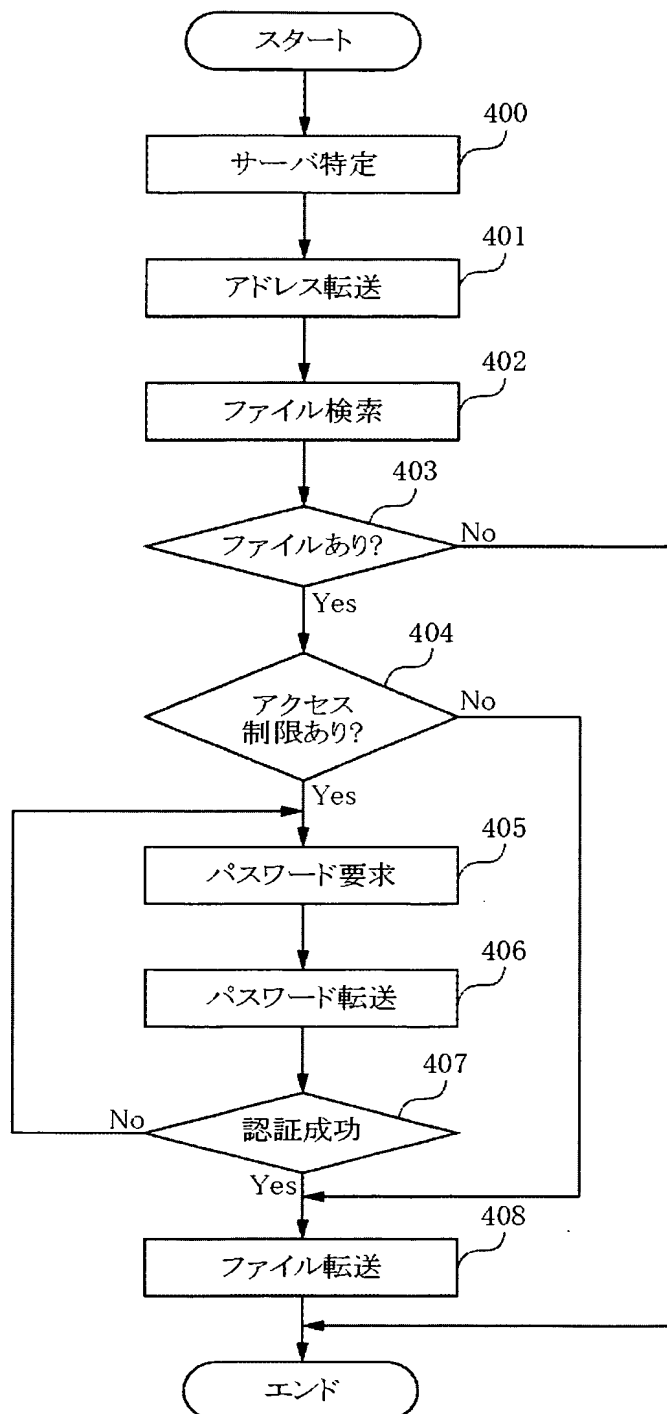
【図 7】



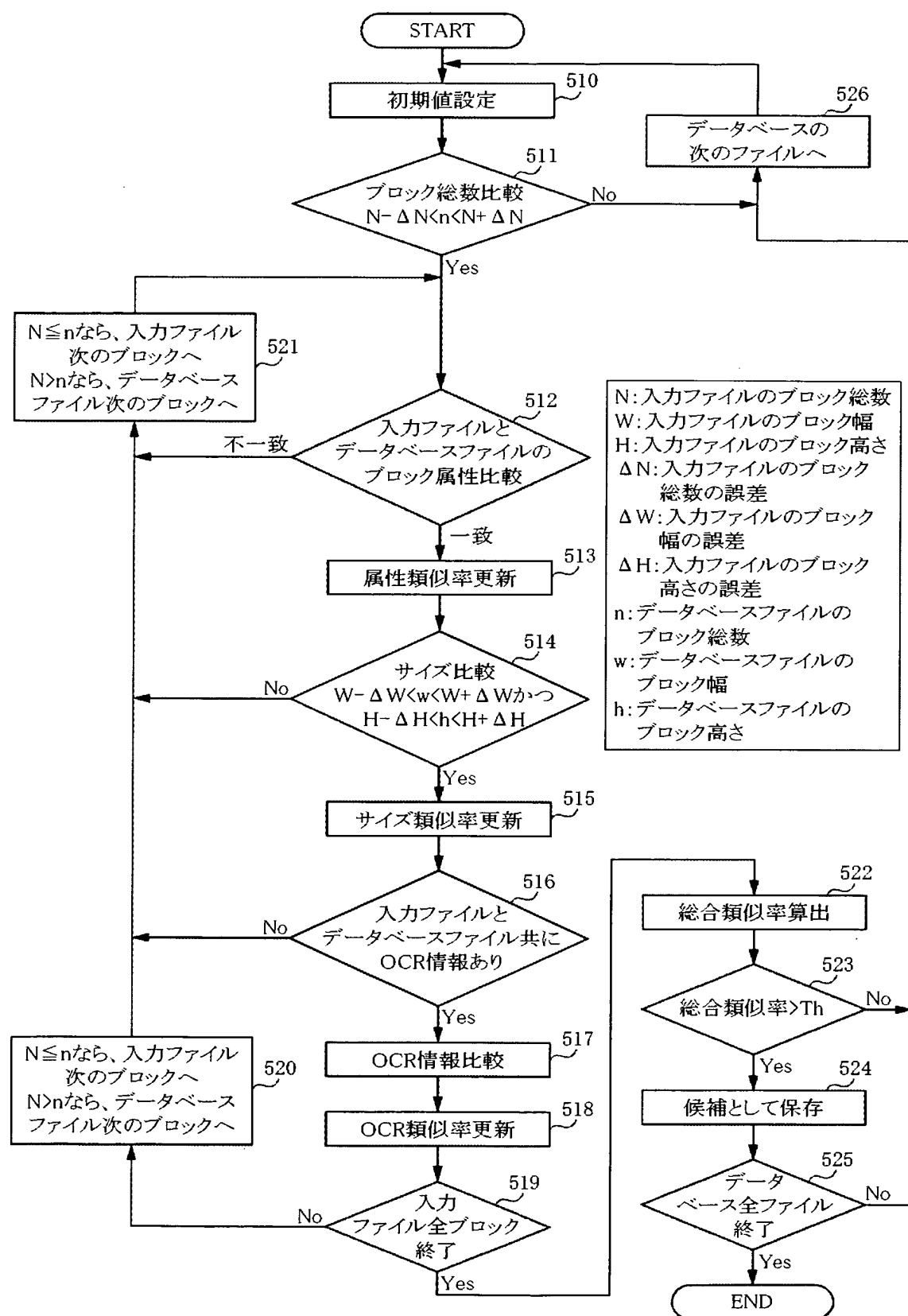
【図 8】



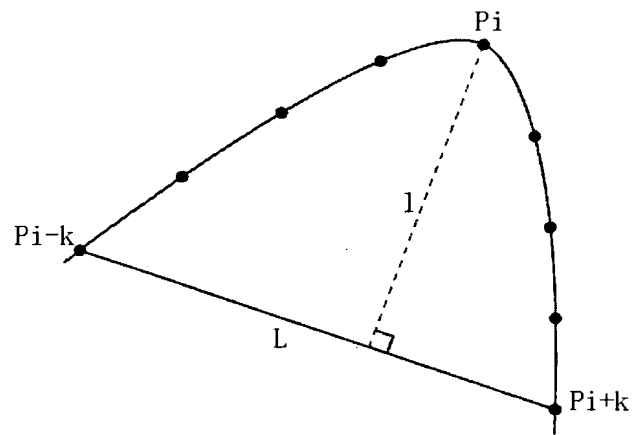
【図 9】



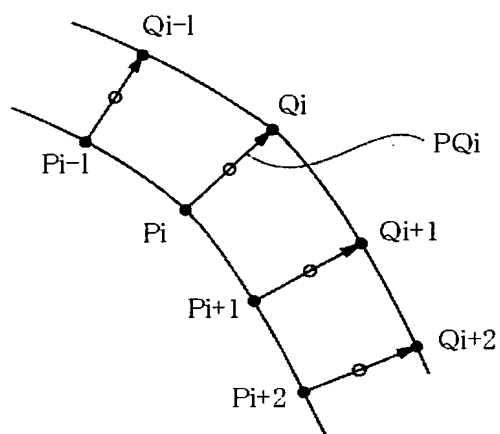
【図 10】



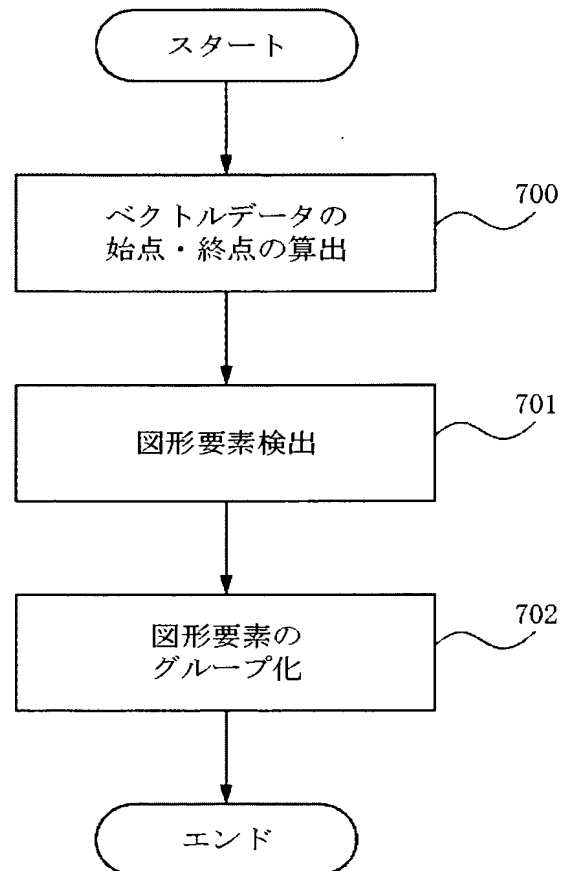
【図 11】



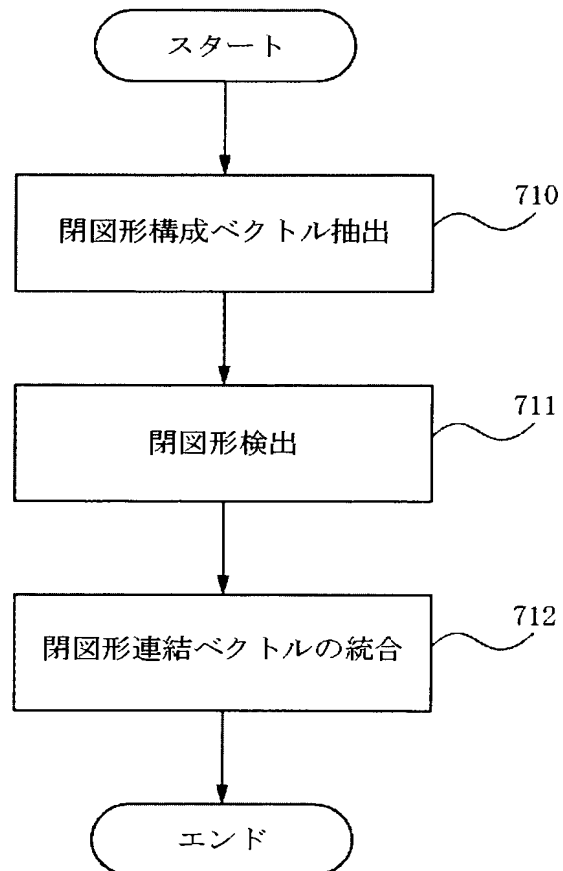
【図 12】



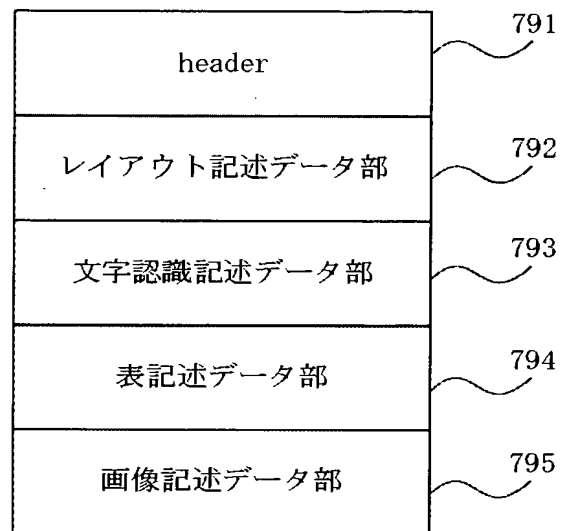
【図 13】



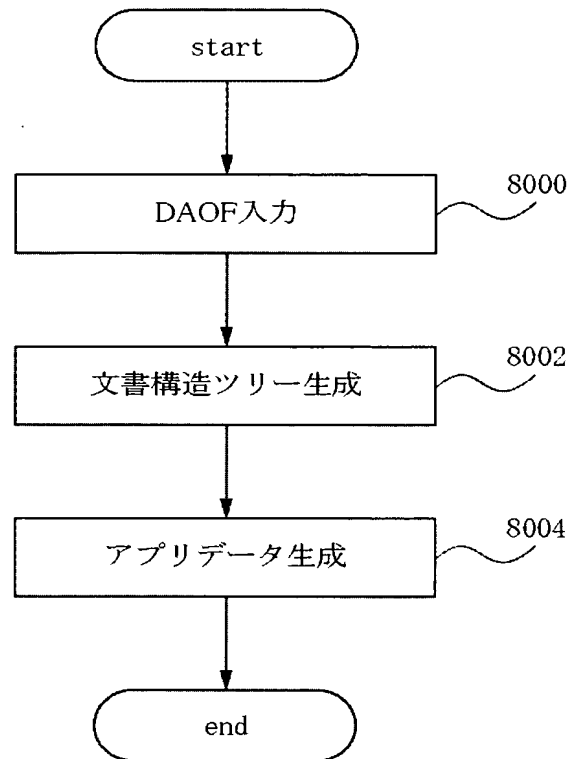
【図 14】



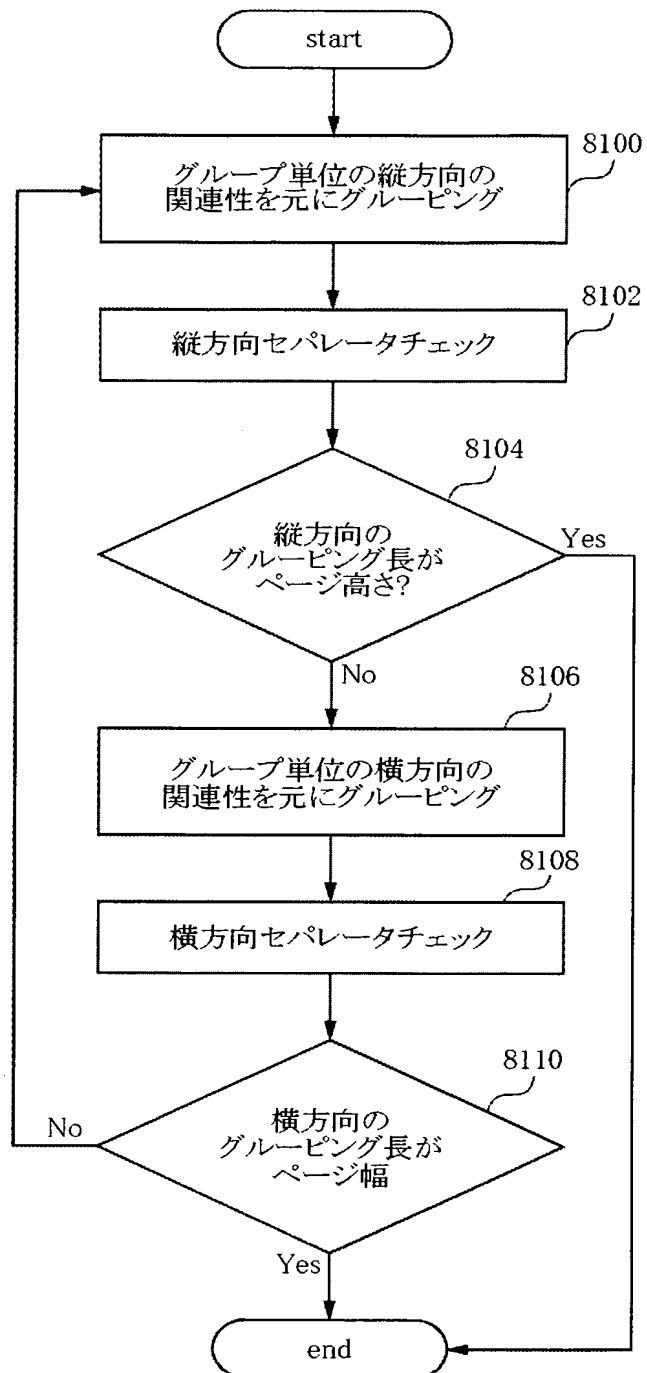
【図 15】



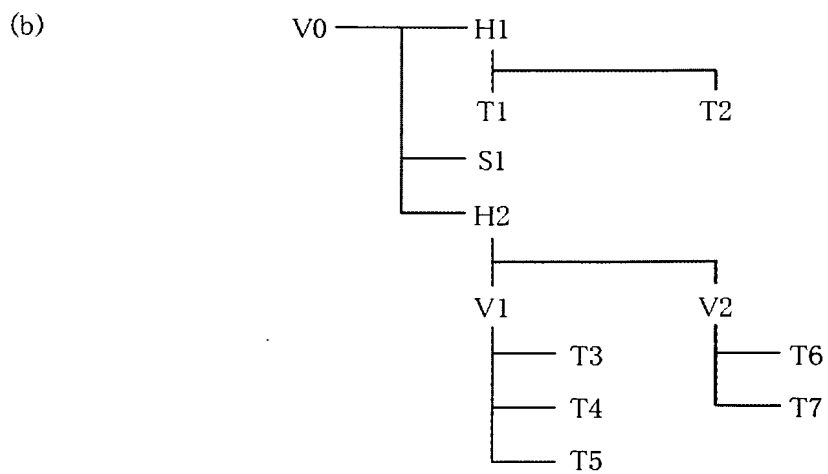
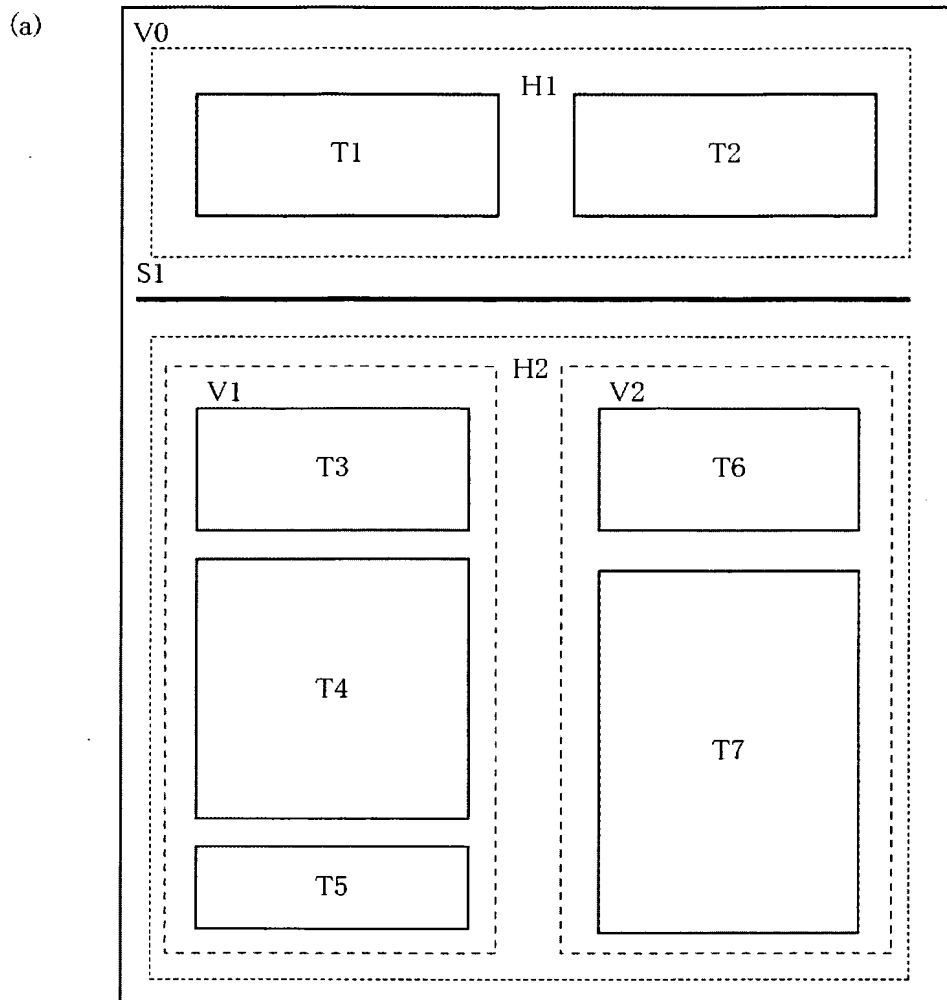
【図 16】



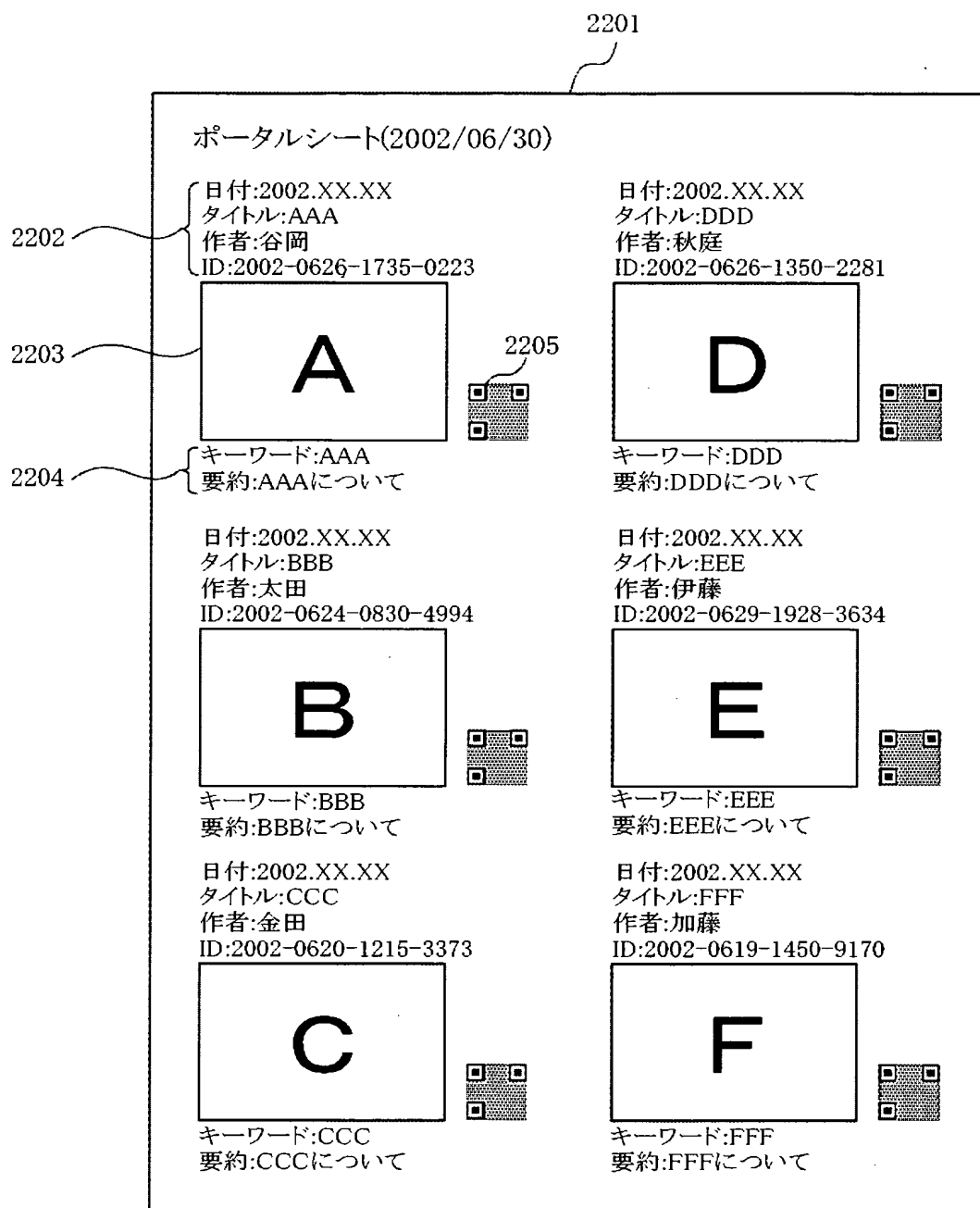
【図 17】



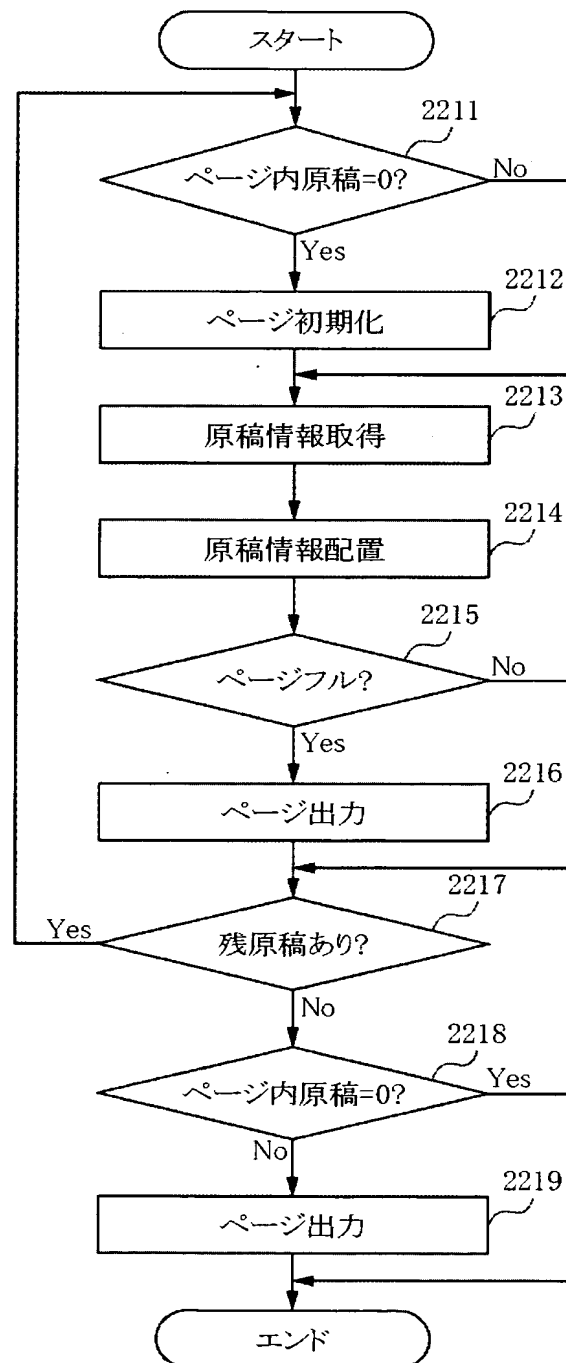
【図 18】



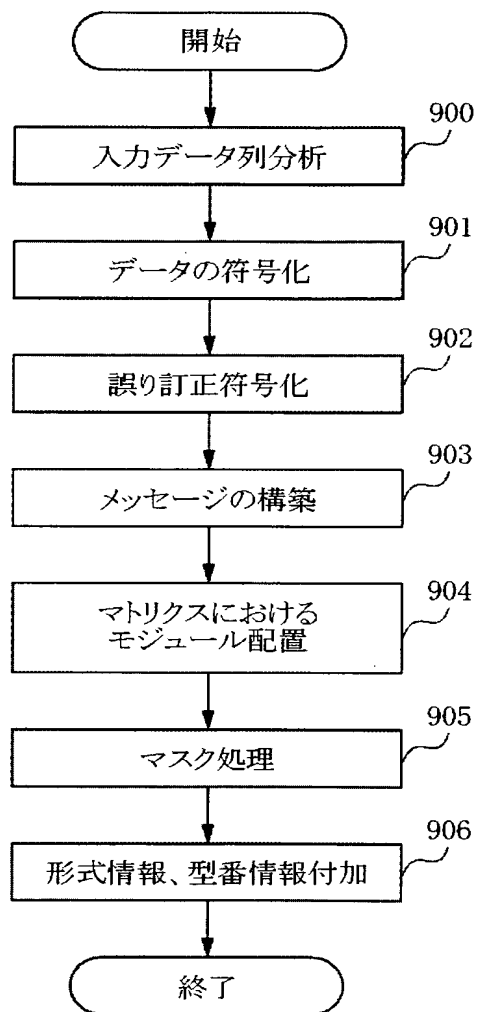
【図 19】



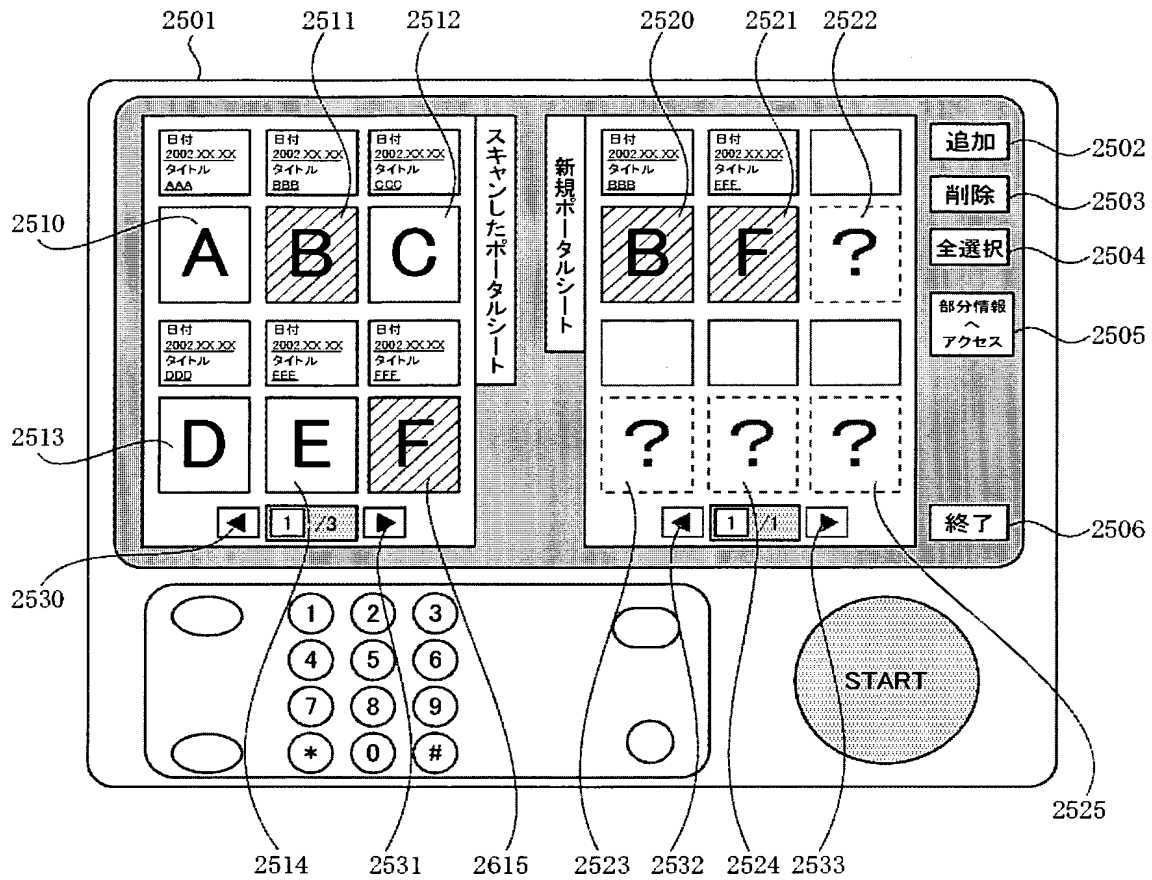
【図 20】



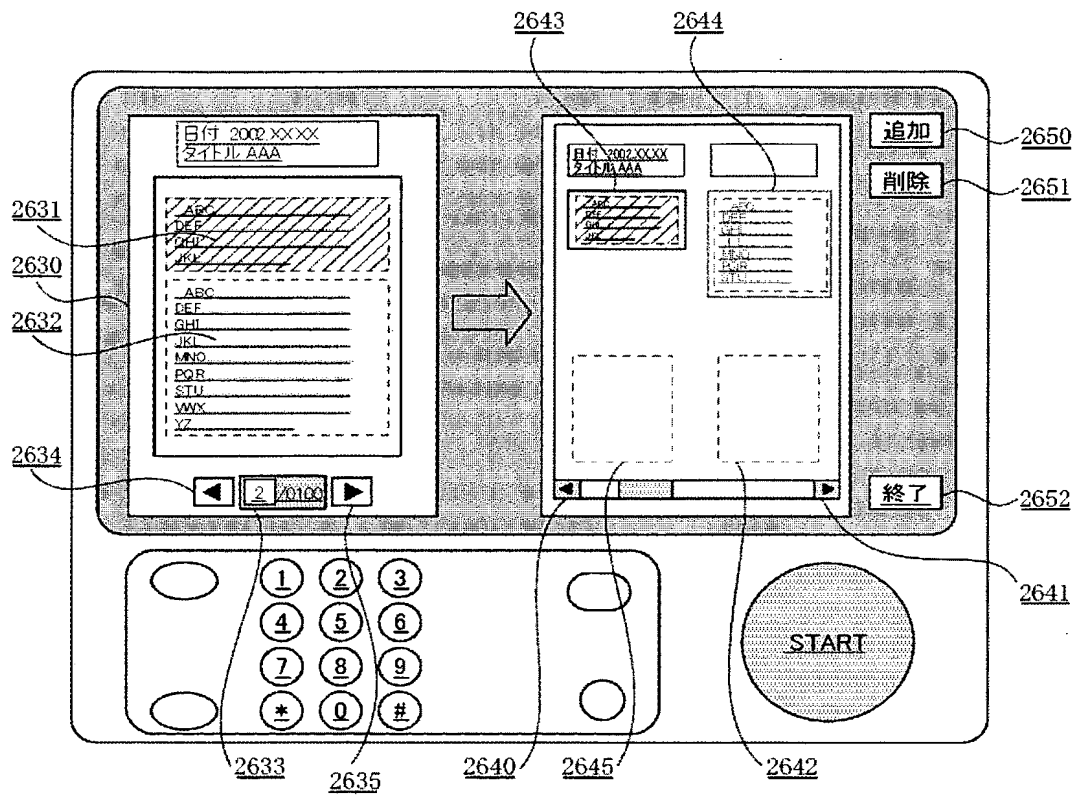
【図 21】



【図 22】



【図 23】



【書類名】 要約書

【要約】

【課題】 画質の劣化も少なく、また記憶容量も少なくて済むようにして格納されたイメージ情報に関して、再利用が容易になるような画像処理方法を提供する。

【解決手段】 入力されたイメージそれぞれに対して、格納手段に格納されているオリジナル電子データを検索し、オリジナル電子データが検索されなかった場合、前記入力されたイメージをベクトルデータ化して電子データとして格納手段に格納し、前記検索ステップでオリジナル電子データが検索された場合において当該検索されたオリジナル電子データに関する情報、あるいは前記検索ステップでオリジナル電子データが検索されなかった場合において前記ベクトル化ステップでベクトル化されて前記格納ステップで格納された電子データに関する情報、の少なくともいずれかを含むシートを生成することにより、再利用が容易になるシートを提供する。

【選択図】 図3

特願 2 0 0 3 - 0 4 4 2 9 9

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 1 0 0 7]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キャノン株式会社